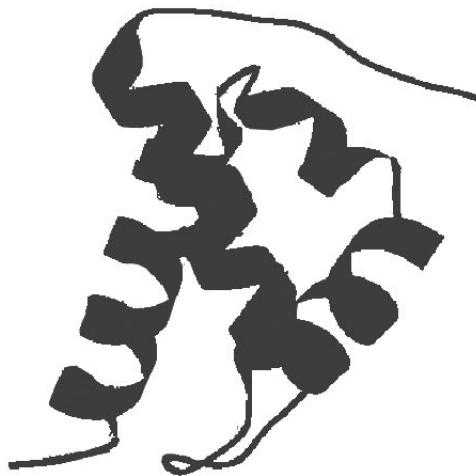


Proceedings of the
Second BioCreative
Challenge
Evaluation Workshop



Proceedings of the //
Second BioCreative
Challenge
Evaluation Workshop



CNIO Centro Nacional de Investigaciones Oncológicas

Melchor Fernández Almagro, 3

28029 Madrid

www.cnio.es

Coordination and edition Lynette Hirschman, Martin Krallinger & Alfonso Valencia

Direction of art and production Bocetocolor SL

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reproduction on microfilms or in any other way, and storage in data banks.

© Fundación CNIO Carlos III, 2007

ISBN 84-933255-6-2

Printed in Spain



Biocreative is supported by a grant from the ESF Programme “Frontiers of Functional Genomics” and by donations from the ENFIN European Commission FP6 Programme NoE LSHG-CT-2005-518254 and from the Spanish National Cancer Research Centre CNIO.





index

- 7** *BioCreative 2. Gene Mention Task*
- 17** *Overview of BioCreative II Gene Normalization*
- 29** *Evaluating the Detection and Ranking of Protein Interaction relevant Articles: the BioCreative Challenge Interaction Article Sub-task (IAS)*
- 41** *Assessment of the Second BioCreative PPI task: Automatic Extraction of Protein-Protein Interactions*
- 55** *Annotating molecular interactions in the MINT database*
- 61** *IntAct - Serving the text-mining community with high quality molecular interaction data*
- 69** *IASL Systems in the Gene Mention Tagging Task and Protein Interaction Article Sub-task*
- 77** *Identifying Gene Mentions by Case-Based Classification*
- 81** *Combined Conditional Random Fields and n-Gram Language Models for Gene Mention Recognition*
- 85** *Tackling the BioCreative2 Gene Mention task with Conditional Random Fields and Syntactic Parsing*
- 89** *Named Entity Recognition with Combinations of Conditional Random Fields*
- 93** *Gene Mention Recognition Using Lexicon Match Based Two-Layer Support Vector Machines*
- 97** *Using Semi-Supervised Techniques to Detect Gene Mentions*
- 101** *BioCreative II Gene Mention Tagging System at IBM Watson*
- 105** *Rich Feature Set, Unification of Bidirectional Parsing and Dictionary Filtering for High F-Score Gene Mention Tagging*
- 109** *High-Recall Gene Mention Recognition by Unification of Multiple Backward Parsing Models*
- 113** *Attribute Analysis in Biomedical Text Classification*
- 119** *Penn/UMass/CHOP Biocreative II systems*
- 125** *Text Detective: Gene/protein annotation tool by Alma Bioinformatics*
- 131** *Peregrine: Lightweight gene name normalization by dictionary lookup*
- 135** *Gene Mention and Gene Normalization Based on Machine Learning and Online Resources*
- 141** *Me and my friends: gene mention normalization with background knowledge*
- 145** *Context-Aware Mapping of Gene Names using Trigrams*
- 149** *ProMiner: Recognition of Human Gene and Protein Names using regularly updated Dictionaries*
- 153** *Human Gene Normalization by an Integrated Approach including Abbreviation Resolution and Disambiguation*
-

-
- 157** *A Hybrid Gene Normalization approach with capability of disambiguation*
- 161** *Exploring Match Scores to Boost Precision of Gene Normalization*
- 165** *Rule-based Gene Normalization with a Statistical and Heuristic Confidence Measure*
- 169** *Automatically Expanded Dictionaries with Exclusion Rules and Support Vector Machine Text Classifiers: Approaches to the BioCreAtivE 2 GN and PPI-IAS Tasks*
- 175** *A Semi-Supervised Approach To Learning Relevant Protein-Protein Interaction Articles*
- 179** *ProtIR prototype: abstract relevance for Protein-Protein Interaction in BioCreAtivE2 Challenge, PPI-IAS subtask*
- 183** *A Term Investigation and Majority Voting for Protein Interaction Article Sub-task 1 (IAS)*
- 187** *Identifying Protein-Protein Interaction Sentences Using Boosting and Kernel Methods*
- 193** *OntoGene in Biocreative II*
- 199** *GeneTeam Site Report for BioCreative II: Customizing a Simple Toolkit for Text Mining in Molecular Biology*
- 209** *AKANE System: Protein-Protein Interaction Pairs in the BioCreAtivE2 Challenge, PPI-IPS subtask*
- 213** *Consensus pattern alignment to find protein-protein interactions in text*
- 217** *Identifying Protein-Protein interactions in Biomedical publications*
- 227** *Integrating knowledge extracted from biomedical literature: normalization and evidence statements for interactions*
- 237** *Mining Physical Protein-Protein Interactions by Exploiting Abundant Features*
- 247** *Uncovering Protein-Protein Interactions in the Bibliome*
- 257** *An integrated approach to concept recognition in biomedical text*
- 273** *Adapting a Relation Extraction Pipeline for the BioCreAtivE II Tasks*
- 287** *Extracting Interacting Protein Pairs and Evidence Sentences by using Dependency Parsing and Machine Learning Techniques*
- 293** *Protein Interaction Sentence Identification by Using Hierarchical Pattern-Based Approach*
- 297** *BioText Report for the Second BioCreAtivE Challenge*
- 307** *LingPipe for 99.99% Recall of Gene Mentions*
- 311** *IASL Systems in the Gene Mention Tagging Task and Protein Interaction Article Sub-task*
- 319** *A Study for Application of Discriminative Models in Biomedical Literature Mining*





BioCreative 2. Gene Mention Task

John Wilbur Larry Smith Lorrie Tanabe
wilbur@ncbi.nlm.nih.gov lsmith@ncbi.nlm.nih.gov tanabe@ncbi.nlm.nih.gov

National Center for Biotechnology Information, Bethesda, Maryland

Abstract

There were 21 participants in the BioCreative 2 Gene Mention Task, with a highest F-score of 87.21. We discuss the statistical significance of these results, and estimate how these systems would perform on alternate corpora. We also demonstrate that by combining the results from all submissions, an F-score of 90.66 is feasible, and furthermore, that the best result makes use of the lowest scoring submissions.

1 Introduction

Finding gene names in scientific text is both important and difficult. It is important because it is needed for tasks such as document retrieval, information extraction, summarization, and automated text mining, reasoning, and discovery. Technically, finding gene names in text is a kind of named entity recognition similar to the tasks of finding person names and company names in newspaper text [2]. However, finding gene names may be significantly harder for several reasons:

1. There are millions of gene names used.
2. New names are created continuously.
3. Authors usually do not use proposed standardized names, which means that the name used depends on preference.
4. Gene names naturally co-occur with other types, such as cell names, that have similar morphology, and even similar context.
5. Expert readers may disagree on which parts of text correspond to a gene name.
6. Unlike companies and individuals, genes are not defined unambiguously. A gene may refer to a specified sequence of DNA base pairs, but that sequence may vary in nonspecific ways, as a result of polymorphism, multiple alleles, translocation, and cross-species analogues.

All of these things make gene name finding a unique and persistent problem. An alternative approach to finding gene names in text, is to decide upon the actual genes that are referenced in a sentence. This is the goal of the gene normalization task [10]. While success in gene normalization to some degree eliminates the need to find explicit gene mentions, it will probably never be the case that gene normalization is more easily achieved. Therefore, the need for finding gene mentions will probably continue into the future.

1.1 Task Description

BioCreative is called a “challenge evaluation” (competition or contest), in which participants are given well defined text-mining or information extraction tasks in the biological domain. Participants are given a common training corpus, and a period of time to develop systems to carry out the task. At a specified time, the participants are then given a test corpus, and a very short period of time in which to apply their systems and return the results to the organizers for evaluation. All submissions are then evaluated according to numerical criteria, specified in advance. The results are then returned to the participants and subsequently made public in a workshop and coordinated publication. The first challenge was carried out in 2003 (with a workshop in 2004) and consisted of a gene mention task, a gene normalization task and a functional annotation task. The current challenge took place in 2006 and the workshop is taking place in 2007. There were three tasks in “BioCreative 2”, called the gene mention (GM), gene normalization (GN) and protein-protein interaction (PPI) tasks. This paper summarizes the performance of the participants in the gene mention task, and also suggests a prospective view of the task.

The BioCreative 2 Gene Mention task builds on the similar task from BioCreative 1. The training corpus for the current task consists mainly of the training and testing corpora from the previous task, and the testing corpus for the current task consists of an additional 5,000 sentences that were held “in reserve” from the previous task. In the time since the previous challenge, the corpus has been reviewed for consistency using a combined automated and manual process. In the previous task, participants were asked to identify gene mentions by giving a range of tokens in the pretokenized sentences of the corpus. In the current corpus, tokenization is not provided, instead participants are asked to identify gene mentions by giving the start and end characters in each sentence. As before, the training set consists of a set of sentences, and to each sentence a set of gene mentions. Each “official” gene mention in a sentence may optionally have alternate boundaries that are judged by human annotators to be essentially equivalent references.

Every substring identified by a run is considered either a true positive or a false positive. If the string matches a gene or alternate in the humanly annotated corpus, it is counted as a true positive with the exception that only one true positive is permitted per gene given in the corpus. If a gene that is given in the corpus does not match any strings nominated by a run, and none of the allowed alternates are matched by a run, then the gene is counted as a false negative. A run is scored by counting the true positives (TP), false positives (FP), and false negatives (FN). Let $T = TP + FN$ denote the total number of genes in the corpus, and let $P = TP + FP$ denote the total number of nominated gene mentions by a run. The evaluation is based on the performance measures p (precision), r (recall), and their harmonic average F

$$p = \frac{TP}{P} \quad r = \frac{TP}{T} \quad F = \left(\frac{p^{-1} + r^{-1}}{2} \right)^{-1} = \frac{TP}{(T+P)/2}$$

Different applications may favor a different weighting between precision and recall, but this is beyond the scope of our analysis. We assume this simple form of F-score in all of our analysis.

Despite being called a ”challenge evaluation”, competition, or contest, there are several reasons to view the results differently. As is pointed out repeatedly in the TREC workshop [8], the ”absolute value of effectiveness measure is not meaningful”, that is, the scores provided are not meaningful outside of the context of the challenge. The F-score is a specific metric, not without controversy, and the value achieved on the corpora of the challenge is no guarantee of performance on other corpora. We will demonstrate how it may be possible to estimate the performance on alternative corpora, but there is no way to determine the accuracy of these estimates. All performance measures have a natural statistical variation, even within the narrow confines of the corpora defined for this task. We will estimate the statistical significance of pairwise comparisons. Finally, runs that score below the median may still give valuable insights into the task, and we will provide some evidence that

this is the case. In short, this competition is not a horse race, but a scientific forum in which the state-of-the-art is advanced through comparison and sharing of ideas.

1.2 Corpus Preparation

In 2003, as part of a project to improve on the AbGene tagger [6], a corpus of 20,000 sentences was selected and annotated for training and testing purposes. As described in [6], a Bayesian classifier was developed to recognize documents that are likely to contain gene names, and it was found that the precision and recall of the tagger was much better for high scoring documents. With this motivation, 10,000 sentences from high scoring documents and 10,000 sentences from low scoring documents were selected and combined to form the 20,000 sentence corpus. The corpus was further subdivided into *train*, *test*, *round1*, and *round2* sets of 5,000 sentences, each of which contained equal numbers of high scoring and low scoring sentences. The *train* and *test* sets were provided as the training set in BioCreative 1, and the *round1* set was used as the final evaluation. With some modifications, the *train*, *test*, and *round1* sets were provided as the training set in BioCreative 2, and the *round2* set was used as the final evaluation.

For BioCreative 2, the entire corpus of 20,000 sentences and approximately 44,500 GENE and ALTGENE annotations, was converted to the MedTag database format [5]. To do this, the original sentence in MEDLINE was located (though a few had been removed from MEDLINE and were replaced with sentences existing at the time). The bibliographic information for each sentence was also determined. The token specifications of all previous annotations were changed to character specifications. And because annotations were no longer limited to preset token boundaries, it was necessary to manually review every annotation to confirm or relocate the annotation boundaries. For example, it became possible to annotate a gene that is hyphenated to another word, the combination of which is not a gene mention.

To improve the consistency of annotation, approximately 1,500 strings (containing 2 or more characters) were found that were annotated as GENE or ALTGENE in one sentence and unannotated in another sentence. These strings occurred in approximately 13,500 mentions, of which 4,300 were GENE annotations, 2,200 were ALTGENE annotations, and 7,000 were unannotated. All of these cases were manually reviewed for accuracy and several corrections were made.

2 Summary of Submitted Runs

The BioCreative 1 gene mention task had 15 participants and each was allowed to submit up to 4 runs, categorized as either closed (no additional lexical resources), or open (no restriction). The BioCreative 2 gene mention task had 21 participants and each team was allowed to submit up to 3 runs. There were no restrictions placed on the submissions. The highest achieved F-score for the BioCreative 1 gene mention task was 82.2 while in the current challenge the highest achieved F-score was 87.2. For the purposes of presenting results, and all further analysis in this paper, only one submission from each of the 21 teams with the highest F-score was considered.

The precision, recall, and F-score for each team, in rank order based on F-score, is shown in Table 1. To compute significance, bootstrap resampling was used on the test corpus. For 10,000 trials, a random sample of 5,000 sentences was selected *with replacement* from the test corpus, and the precision, recall, and F-score was computed using these sentences for each of the 21 submissions. For each pair of submissions, say *A* and *B*, the proportion of times in these 10,000 trials that the F-score of *A* exceeded the F-score of *B* was noted, and we label that pair statistically significant if this proportion is greater than 95%. Significant differences are shown in Table 1. One can see that the top 3 F-scores did not have statistically significant differences. Also, the top 6 F-scores are all statistically significant compared to the remaining scores, and so on. Every pair of F-scores ($\times 100$) that differed by approximately 1.23 or more was significant, and every pair of F-scores that

rank	BioCreative					MEDLINE			Trans. Factors		
	p	r	F	signif	% alt	p	r	F	p	r	F
1	88.48	85.97	87.21	4-21	32.48	80.06	83.62	81.80	90.15	86.57	88.32
2	89.30	84.49	86.83	6-21	14.02	83.21	81.14	82.16	90.52	85.31	87.84
3	84.93	88.28	86.57	6-21	14.08	76.53	85.22	80.64	86.77	89.02	87.88
4	87.27	85.41	86.33	7-21	31.77	79.93	82.78	81.33	88.80	85.96	87.36
5	85.77	86.80	86.28	7-21	16.67	73.53	83.84	78.35	88.45	87.47	87.96
6	82.71	89.32	85.89	7-21	16.02	69.78	87.85	77.78	85.72	89.66	87.65
7	86.97	82.55	84.70	8-21	14.83	78.62	78.88	78.75	88.75	83.43	86.01
8	84.35	81.39	82.85	10-21	14.57	74.42	77.30	75.83	86.60	82.40	84.45
9	86.28	79.66	82.84	10-21	14.55	79.33	75.97	77.61	87.77	80.48	83.97
10	85.22	78.44	81.69	12-21	33.02	75.82	74.73	75.27	87.16	79.29	83.04
11	85.54	76.83	80.95	12-21	19.76	75.64	74.07	74.84	87.73	77.48	82.29
12	72.95	88.49	79.97	14-21	16.82	50.75	88.31	64.46	79.26	88.46	83.61
13	92.67	68.91	79.05	15-21	19.73	89.88	64.39	75.02	93.25	70.10	80.04
14	88.83	69.70	78.11	16-21	37.05	82.44	64.39	72.30	90.05	71.20	79.52
15	80.46	73.61	76.88	17-21	20.43	71.92	70.85	71.38	82.10	74.08	77.89
16	82.28	71.08	76.27	18-21	16.80	73.40	67.33	70.23	84.26	71.95	77.62
17	84.32	68.57	75.63	18-21	34.02	80.40	64.61	71.64	85.01	69.39	76.41
18	71.68	62.33	66.68	19-21	28.23	54.16	61.33	57.52	75.99	62.33	68.49
19	65.83	61.55	63.62	20, 21	27.23	49.98	55.95	52.79	69.39	62.78	65.92
20	60.56	64.11	62.29	21	31.71	39.30	62.45	48.24	66.98	64.54	65.74
21	50.09	46.12	48.02	-	28.46	36.71	43.86	39.97	53.44	46.42	49.68

Table 1: In the left panel, under “BioCreative”, the precision, recall, and F-score for the best submitted run from each of 21 participants, sorted by F-score. Each team has an F-score that has a statistically significant comparison ($p < 0.05$) with the teams indicated in the *signif* column. The column labeled % alt is the percentage of true positives in the submission that matched an ALTGENE annotation. The panels under “MEDLINE” and “TransFactor” are the precision, recall and F-score after reweighting sentences for MEDLINE and a “human blood transcription factors” query, respectively (see text).

differed by approximately 0.35 or less was insignificant.

Table 1 also shows the alternates in each run as a percentage of the corresponding true positives, which varies from about 15 to 30%. It is interesting to observe that the number of alternates in a run is not predictive of the score, as the top 3 runs represented both extremes. Nevertheless, there was an overall negative correlation of -0.40, and it could be hypothesized that methods which were less effective at learning the boundaries of the primary gene mentions were still able to get close enough to match alternatives, resulting in a higher representation of alternates among their true positives.

If the percentage of alternates among true positives is denoted by α , then the F-score that obtains after omitting the alternates is

$$F^* = F \frac{1 - \alpha}{1 - F\alpha/2} = F(1 - \alpha)(1 + F\alpha/2 + (F\alpha/2)^2 + \dots).$$

With α in the observed range, removing the alternates reduces the F-score by a multiple ranging from $1/2$ to $3/4$ of α , with the greater reduction occurring for lower scoring runs.

Along with each submitted run, each team was required to submit answers to a list of questions which are paraphrased in Figure 1. We read the submitted answers and developed a list of mentioned features, which are shown in Table 2. That table also shows the total number of features

How would you summarize your overall approach?

List training data that you used in addition to the training data provided.

List machine learning techniques used.

List NLP techniques used.

List Bio-NLP techniques used.

List external lexical resources used, such as dictionaries and ontologies.

Figure 1: Paraphrase of the questionnaire required from each participant in the Gene Mention task.

Techniques	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	Tot
SVM	*	*											*									3
CRF		*	*	*	*	*	*	*					*		*	*			*			11
Merge	*	*						*			*	*										5
Online learning				*																		1
n-gram							*															1
MaxEnt								*														1
HMM									*													1
Manual rules														*			*					2
Case based														*								1
C4.5																	*					1
NLP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
POS tagger	*	*	*		*	*	*		*	*	*	*	*	*	*	*						12
NP chunker	*							*														3
Paren matching	*						*															2
Stemming	*	*						*		*							*					5
Bidirectional	*	*																				2
Abbreviations			*					*														3
LSA			*																			1
Character based				*									*									2
Tokenization					*											*						2
Parser						*											*					3
Systems	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
Mallet	*	*	*		*										*							5
GENIA tagger			*		*									*								4
LingPipe					*									*								2
Abner											*					*						2
TnT										*												1
MedPost												*	*									2
Data	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
Medline	*			*	*	*																4
Mesh	*					*																2
LocusLink	*																					1
SwissProt	*																					1
HUGO	*			*	*									*								4
Other lists				*	*			*										*				4
ALTGENE				*					*													2
AbGene List				*																		1
Biothesaurus						*																1
UMLS						*																1
RefSeq							*															1
MedPost							*															1
EntrezGene										*					*			*				3
Genia															*							1
Uniprot															*							1
WordNet																*						1
Totals	8	7	7	10	6	11	8	4	4	3	5	3	6	3	4	5	5	1	2	3	2	

Table 2: Features mentioned in system questionnaires, as interpreted by the authors, for the best run from each team. Column headings are the F-score rank. The last column is the number of teams that mention a feature, and the last row is the number of features mentioned by each team.

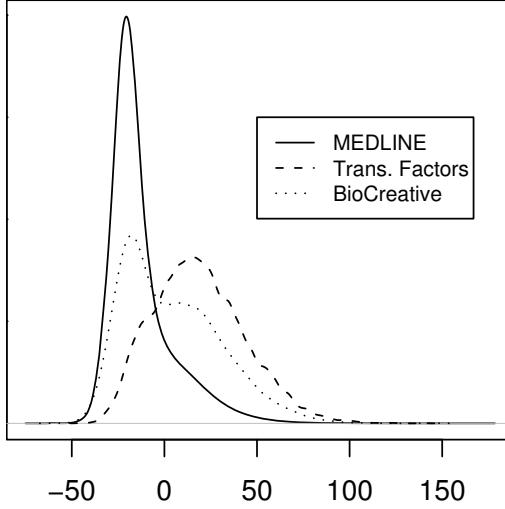


Figure 2: The distributions of gene indicator scores for the sentences of MEDLINE, the sentences in the query “Human Blood Transcription Factors” and the 5,000 sentences comprising the BioCreative 2 evaluation set. The distribution of scores for the BioCreative 2 sentences reflects the origin of the corpus as an equal mixture of high scoring and low scoring sentences.

mentioned by each team, and the total number of teams mentioning each feature (the rows are sorted in each group according to the highest ranking team mentioning the feature). Because this questionnaire was not a controlled study, it is not possible to draw definite conclusions as to the relative effectiveness of the techniques. However, it is interesting to note that the number of features mentioned by a team has a significant correlation of -0.757 with the rank of the submission.

3 Estimated Performance on Alternate Corpora

As noted in Section 1.2, the corpus provided for training and testing was selected from MEDLINE so as to equally represent sentences likely to contain gene names and sentences not likely to contain gene names. Because of this selection bias, the performance measures obtained in this evaluation do not directly predict the system performance in any other situation. Nevertheless, by weighting the sentences appropriately, it is possible to estimate a system’s performance on corpora with a different distribution of sentences.

Based on the selection bias in the original 20,000 sentences, we used an updated system for scoring sentences to indicate whether they are likely to contain a gene name, and we used this score to obtain weights for alternative corpora. Suppose it is desired to estimate the performance of a system on a given alternate corpus. If f_t is an estimate of the probability density function for the scores in the test set, and f_a is the probability density function for the scores in the alternate corpus, then sentence number i from the test set with score s_i should have weight $w(i) = f_a(s_i)/f_t(s_i)$ in the alternate corpus. Then, if sentence number i contains $TP(i)$ true positive gene annotations, $FP(i)$ false positives and $FN(i)$ false negatives, then the weighted performance on the alternate collection is computed using

$$TP' = \sum_i w(i)TP(i) \quad FP' = \sum_i w(i)FP(i) \quad FN' = \sum_i w(i)FN(i)$$

To estimate the densities f_t and f_a we computed the scores for all of the sentences of the col-

lections and then applied the *density* function using the R 2.2.1 statistical program (with spline interpolation).

We carried out this weighting for random sentences selected from MEDLINE and for sentences selected as the result of a query for human blood transcription factors. The distribution of the gene score for the sentences in the BioCreative test corpus is shown in Figure 2, along with the distribution of scores of random sentences from MEDLINE and the sentences from the PubMed query

```
”Transcription Factors” [MeSH]  
AND ”Blood Cells” [MeSH]  
AND Humans [MeSH]
```

which returns 9,003 abstracts. One can see that the BioCreative test corpus has a greater representation of high scoring sentences than the MEDLINE corpus, as does the human blood transcription factor corpus. The computed precision, recall, and F-scores for each team for the alternate corpora are shown in Table 1. Whether a system performs better or worse on a collection roughly depends on the difference in its performance on high scoring and low scoring sentences. Note that the estimated F-scores for random MEDLINE is generally lower than the scores on the BioCreative 2 evaluation, while the estimated F-scores for the human blood transcription factor set are generally higher.

4 Combined Performance

We wanted to know if it is possible to improve on the best scores obtained in this workshop. To do this, we used machine learning to predict gene mentions using all of the submitted runs as feature data.

In order to simulate what might result if all of the methods were combined into a single system, we extracted features from the submitted runs. By holding out 25 sentences at a time, and training on the remaining 4,975 sentences, we could apply the result to the held out set and then merge all of the results to obtain a single “fusion” run for all 5,000 sentences.

For each candidate, which is defined by a particular start and end offset within a sentence, the features described in Figure 3 were generated. We used two different machine learning techniques with this feature data, boosted decision trees, and conditional random fields.

For boosted decision trees, the training set consisted of all candidates whose starting and ending offsets coincided with a nominated string from at least one team (but the starting and ending offsets need not both be nominated by the same team). Each character of a candidate was also required to overlap a nominated string from at least one team. This meant that every candidate had at least one “nom” feature from Figure 3. Each candidate was further marked as a “positive” depending on whether it appeared exactly as a gene or alternate gene mention, and all other candidates were marked as a “negative”. A boosted decision tree algorithm [4, 1] was applied to this data set (holding out 25 sentences at a time, as mentioned above) to learn which candidate is a positive. Each tree was allowed to have a depth of 5 and boosting was repeated 1,000 times. The induced set of decision trees was applied to the held-out set of 25 sentences to obtain gene mentions for them. Where gene mentions overlap, only the gene mention with the highest score is retained, so that the final result does not contain any overlapping gene mentions. We repeated this training using only nomination features, only word features, and combined nomination and word features. The results are shown in Table 3, and the nomination features combined with words performed best with an F-score of 90.50. As this is 3.29 greater than the highest F-score obtained by an individual team, the difference is statistically significant.

We also used conditional random fields (with gaussian prior) to learn gene mention [3]. Each sentence was tokenized and each token was marked as being positive or negative depending on

$not(T)$	Team T did not nominate any gene mention that overlaps with this candidate.
$nom(T)$	Team T nominated a gene mention that overlaps with this candidate.
$noms(T, S)$	Team T nominated a gene mention that overlaps with this candidate, and that starts before ($S = -1$), starts after ($S = 1$) or coincides with the start of this candidate ($S = 0$).
$nome(T, E)$	Team T nominated a gene mention that overlaps with this candidate, and that ends before ($E = -1$), ends after ($E = 1$) or coincides with the end of this candidate ($E = 0$).
$nom(T, S, E)$	Team T nominated a gene mention with S and E as above.
$noms(S)$	Some team nominated a gene mention with S as above.
$nome(E)$	Some team nominated a gene mention with E as above.
$word(W)$	Word W occurs in the candidate.
$firstword(W)$	Word W is the first word of this candidate.
$lastword(W)$	Word W is the last word of this candidate.
$context(P, W)$	Word W at position P relative to this candidate. The possible values for P are $2, -1, 1, 2$.

Figure 3: The features generated for each candidate gene mention, based on the submitted runs.

whether it was part of an annotated gene (alternates were not used in this approach). The features described in Figure 3 were generated for each token, in which, for the purposes of generating features, each token is treated as a candidate. By holding out 25 sentences at a time, the CRF was trained on the remaining 4,975 sentences (the gaussian prior defined in [3] was taken to be $1/2\sigma^2 = 300$). The trained CRF was then applied to tag the 25 sentences, and any sequence of consecutive positive labels were combined into a single gene mention. The results from each set of 25 sentences were combined to form a single run. The result, shown in Table 3 was an F-score of 90.66. This is slightly higher than the result obtained using boosted decision trees (with nomination and word features), but the difference is not statistically significant.

A question of interest to us is whether the alternate annotations could be used in machine learning to improve performance in the gene mention task. There were some teams that did train

Exp	Method	<i>p</i>	<i>r</i>	<i>F</i>	<i>signif</i>	% alt
A	CRF noalt, nom and word	92.55	88.85	90.66	1-21, C-F	13.62
B	BDT nom and word	92.21	88.85	90.50	1-21, C-F	25.67
C	BDT nom and word, top 10 teams	91.18	87.68	89.40	1-21, E, F	23.37
D	BDT nom only	90.92	87.73	89.29	1-21, E, F	25.42
E	BDT noalt, nom and word	92.42	81.65	86.70	7-21, F	9.58
F	BDT word only	71.65	61.87	66.40	19-21	37.07

Table 3: The precision, recall, and F-score of machine learning experiments to learn gene mentions using the data extracted from all submitted runs as features. Method column: BDT = boosted decision trees, CRF = conditional random fields, nom = all nomination features, word = words of candidate, noalt = alternate gene data not used. The column *signif* indicates the ranks of runs for which there was a significant difference, and the letters indicate the machine learning experiments for which there was a significant difference. The column % alt gives the percentage of alternate gene mentions among the resulting true positives.

with alternates, but the data from individual runs is not sufficient to settle the issue. Given that the boosted decision tree result, which uses alternates, is about the same as the conditional random field result, we might conclude that training with alternates does not make the task significantly easier. We therefore trained with boosted decision trees, marking candidates as positive only if they appear as gene annotations, *i.e.* ignoring alternates. The result was an F-score of 86.70, which is a statistically significant difference with the result 90.50 obtained by training in the same way with alternates positive. Training with alternates generated true positives that contained 25.67% alternates, while training without alternates generated true positives containing only 9.58% alternates.

We believed that the results from the lowest scoring teams, if used appropriately, could contribute useful information towards identifying gene mentions. To test the hypothesis, we trained with boosted decision trees using word features plus all nomination features from teams ranked 1 through 10 only. The result gave an F-score of 89.40, which is significantly lower than the 90.50 obtained when features from teams with ranks 11 through 21 were included. This confirms the importance of results from teams with lower individual performance. We note, for example, that the lowest ranking team obtained 8 true positives that were not obtained by any other run.

5 Discussion

The submission data can be used as a source for exploring the consistency and accuracy of corpus annotations. There were no false positives common to all 21 submissions, but there were 2 that were common to 17 submissions, for the names *GH* and *FAK*, both of which should have been annotated as true. There are more of these false positives with less than 17 common submissions that deserve further review. We also found 34 gene mentions that were false negatives in all 21 submissions, but all of these were found to be correctly annotated in the corpus. Mentions with a high false negative rate may be clues to difficult or under-represented gene mentions, and studying these may give some guidance to future systems developers. As much as we would like to increase the representation of “difficult” gene mentions, this may be infeasible because it is likely that they obey a Zipf-like distribution: there are as many uniquely difficult gene mentions as there are common and easy ones.

It can be argued that the difficulty experienced by human annotators in reaching mutual agreement directly limits the performance of automated systems, and this can be influenced by the clarity of the annotation guidelines. It has been pointed out that the guidelines for annotating genes are surprisingly short and simple given the complex guidelines for annotating named entities in news wires [2]. However, a gene is a scientific concept, and it is only reasonable to rely on domain experts to recognize and annotate gene mentions. Thus, the gene annotation guidelines can be conveyed by reference to a body of knowledge shared by individuals with experience and training in molecular biology, and it is not feasible to give a complete specification for gene annotation that does not rely on this extensive background knowledge. Nevertheless, we believe that some improvement could be achieved by documenting current annotation decisions for difficult and ambiguous gene mentions.

The highest F-score obtained on the BioCreative 2 evaluation is 87.21, and we have shown that by combining the efforts of all systems it is possible to achieve an F-score of 90.66, a significant improvement. This proves that future systems should be able to achieve improved performance. Though this F-score is only relevant to the BioCreative 2 test corpus, it is feasible, as illustrated here, to estimate the performance of future systems on the current corpus, and thus to measure the improvement in future systems' performance. We are also optimistic that, through a combination of refining the corpus for annotation consistency and improving systems design through collaboration, even greater improvements in performance are achievable.

References

- [1] Carreras, X and Marquez, L. (2001) *Boosting trees for anti-spam email filtering*. In RANLP2001. Tzgov Chark, Bulgaria.
- [2] Hirschman, L and Chinchor, N. (1997) *Muc-7 named entity task definition*. In Proceedings of the 7th Message Understanding Conference.
- [3] McCallum, A. (2003) *Efficiently inducing features of conditional random fields*. In Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03).
- [4] Schapire, RE and Singer, Y. (1999) *Improved boosting algorithms using confidence-rated predictions*. Machine Learning, 1999. 37(3): p. 297-336.
- [5] Smith LH, Tanabe L, Rindflesch T and Wilbur WJ. (2005) *MedTag: A Collection of Biomedical Annotations*. In Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pp. 32-37. Detroit, June 2005.
- [6] Tanabe, L, and Wilbur, WJ. (2002) *Tagging Gene and Protein Names in Biomedical Text*. Bioinformatics, 18:1124-1132, 2002.
- [7] Tanabe L, Xie N, Thom LH, Matten W, Wilbur WJ. (2005) *GENETAG: A Tagged Corpus for Gene/Protein Named Entity Recognition*. BMC Bioinformatics 6(Suppl 1):S3.
- [8] Voorhees, EM. (2005) *Overview of TREC 2005*. NIST Special Publication 500-266.
- [9] Yeh, AS, Morgan, A, Colosimo, M, Hirschman, L. (2005) *BioCreAtIvE task 1A: gene mention finding evaluation*. BMC Bioinformatics 6(Suppl 1):S2.
- [10] *BioCreative 2: Gene Normalization Task*. These proceedings.



Overview of BioCreative II Gene Normalization

Alexander A. Morgan¹
`alexmo@stanford.edu`

Lynette Hirschman²
`lynnette@mitre.org`

¹Biomedical Informatics, Stanford University, Stanford, CA 94305, USA

²Information Technology Center, The MITRE Corporation, Bedford, MA, 01730, USA

Abstract

Background

Our goal in BioCreative has been to assess the state of the art in text mining, with emphasis on applications that reflect real biological applications, e.g., the curation process for model organism databases. This paper summarizes the BioCreative II Gene Normalization task, whose goal is to produce the list of unique gene identifiers for the human genes and gene products mentioned in sets of MEDLINE abstracts. We prepared a training set of 281 human annotated documents and a test set of 262 documents. We made these available to the participants, along with a lexicon of gene identifiers and the corresponding names and gene symbols, as well as a set of 5,000 partially annotated abstracts as additional “noisy” training data. System results were computed by automatic comparison to a gold standard created by expert annotators; where the majority of system results differed from the “gold standard,” these results were rechecked, and the gold standard revised.

Results

Twenty groups fielded between one and three runs for the test data for a total of 54 runs. Three systems had F-measures in the 0.80-0.81 range, and the top 6 systems differed by only 0.38 points of F-measure. The top recall score was 0.875 (at the expense of precision at 0.496, for an F-measure of 0.632), and six teams had recall scores of over 0.80, including the system with the top F-measure (recall of 0.833, and precision of 0.789).

Conclusion

This assessment demonstrates that multiple groups were able to perform the mapping of text mentions to gene identifiers with high accuracy. Overall, 9 out of 20 groups had a run that achieved an F-measure of 0.75 or better, indicating a significant advance in the state of the art for gene normalization.

Keywords: text mining, gene normalization, information extraction, BioCreative

1 Introduction

The goal of the Gene Normalization (GN) task is to identify the unique EntrezGene identifiers of human genes and proteins mentioned in a collection of abstracts taken from MEDLINE. This task has been inspired by a step in the typical curation pipeline for model organism databases. Once an article has been selected for curation (as in the Interaction Article Subtask for Protein-Protein Interaction BioCreative task), the next step is for a curator to list the relevant genes or proteins mentioned in the article. In the real curation process, the curator generally (although not always) curates from the full text of the articles, and identifies only particular kinds of genes of interest (e.g., only genes for a specific organism or only genes that have experimental evidence for their function). However the GN task for BioCreative has been simplified in these two respects: we use freely available abstracts from MEDLINE, rather than full text articles; and all (and only) human genes/proteins mentioned in the abstract are associated with an EntrezGene identifier.

The GN task was also carried out as part of the first BioCreative [1, 2], where the focus was on extraction of

unique gene identifiers for three sets of abstracts from the fly, mouse and yeast model organism databases. For BioCreative II we chose to focus on human gene and protein names, motivated in part by our desire to provide alignment with the protein-protein interaction (PPI) task. In contrast with genomic data for fly, mouse and yeast, data for the human genome is not organized into a single model organism database, which made collection of resources somewhat more complicated. However, we used a very similar approach in that we identified a high accuracy human-annotated data set from genes annotated by the GOA team at EBI. Our goal was to provide small carefully (and completely) annotated training and blind test sets, and a much larger number of abstracts as noisy (incompletely annotated) training data, as described in [3]. We used the *gene_association.goa_human* file (<http://www.geneontology.org/>) downloaded on 10 October 2005 to provide 11,073 PubMed identifiers (and 10,730 abstracts) associated with journal articles likely to have mentions of human genes and proteins. We then used the file *gene2pubmed* obtained from NCBI (<ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/>) on 21 October 2005, along with the GO annotations, to create the automatic/noisy annotations in the 5,000 abstracts set aside as a noisy training set as described [2]. We selected our abstracts for expert annotation from the 5,730 remaining abstracts. The expert annotated training set consisted of 281 documents, and the blind test had 262 documents each with a “gold standard” annotation consisting of the list of unique EntrezGene identifiers for the human proteins in the abstract.

2 Results

Each team was allowed to submit up to three runs. Overall, we received a total of 54 runs from 20 participating teams. For each run, we computed the results based on a simple matching of gene identifiers for an abstract against the gold standard. Identifiers that matched the answer key constitute true positives (TP), identifiers that did not match were false positives (FP), and gold standard identifiers that were not matched were false negatives (FN). Recall, precision and F-measure were computed in the usual way:

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN}) \quad \text{Precision} = \text{TP}/(\text{TP}+\text{FP}) \quad \text{F-measure} = 2*\text{P}*\text{R}/(\text{P}+\text{R})$$

We computed two sets of results: the micro-averaged results, which pooled the results across all documents to compute the total recall, precision and F-measure; and the macro-averaged results, which average the score per document to compute the F-measure. The macro-averaged score was used to determine statistical significance between results, using a two-sided t-test. Table 1 shows the results of the top scoring run for each team; it includes the recall, precision and F-measure for the best (micro-averaged) run of each system. In addition, the table shows the macro-averaged F-measure and rank, as well as the rank of the systems that had a significant different in performance (at the 0.10 level for a one-sided t-test). The ranking of the top seven systems did not change between the macro-averaged scores and the micro-averaged scores. In two cases, the top-scoring run for the system changed for the macro-averaged scores.

Figure 1 shows the micro-averaged results as a scatter plot of precision vs. recall. This plot includes all 54 runs. The highest recall reported on an official run (T42_2) was 0.875, with a precision of 0.496, and an F-measure of 0.633. Several systems reported even higher recall in subsequent experiments on the data (e.g., Team 34 reported a recall of 0.91 at a precision of 0.38). Overall, the system performance clustered into several groups – the top 6 systems were separated by 0.038 points of F-measure (0.811 to 0.773). This underscores the point that there are an increasing number of high-performing systems, compared to the last BioCreative.

Team/ Run	Recall	Precision	F-measure Micro-Avg	Rank	F-measure	Rank Macro	Signif Range
			Micro		Macro-Avg		
T042_1	0.833	0.789	0.810	1	0.811	1	3-20
T034_1	0.815	0.792	0.804	2	0.782	2	8-20
T013_1	0.768	0.833	0.799	3	0.779	3	8-20
T004_1	0.734	0.841	0.784	4	#0.777	4	8-20
T109_1	0.824	0.743	0.781	5	0.775	5	8-20
T104_1	0.743	0.807	0.774	6	0.773	6	9-20
T101_2	0.743	0.801	0.771	7	0.755	7	10-20
T107_1	0.740	0.784	0.761	8	0.739	*9	12-20
T113_2	0.761	0.752	0.756	9	0.745	*8	11-20
T108_3	0.749	0.726	0.737	10	0.724	10	13-20
T007_2	0.703	0.746	0.724	11	0.694	*12	16-20
T017_1	0.708	0.720	0.714	12	#0.710	*11	15-20
T110_1	0.629	0.783	0.698	13	0.685	*14	16-20
T111_3	0.664	0.717	0.689	14	0.664	*15	17-20
T030_1	0.661	0.716	0.687	15	0.649	*16	17-20
T006_2	0.606	0.767	0.677	16	0.686	*13	19-20
T036_1	0.713	0.520	0.602	17	0.595	17	19-20
T014_1	0.485	0.762	0.593	18	0.584	18	20
T102_3	0.790	0.425	0.552	19	0.559	19	20
T058_2	0.415	0.375	0.394	20	0.398	20	

Table 1: Recall, precision and F-measure for best GN run per team, including both micro-average and rank, macro-average and rank, and significance based on macro-averaged score distributions. Asterisks indicate that rank for micro- and macro-average are different; ‡ indicates that a different run was used as the high-scoring run in macro- vs. micro-averaged results.

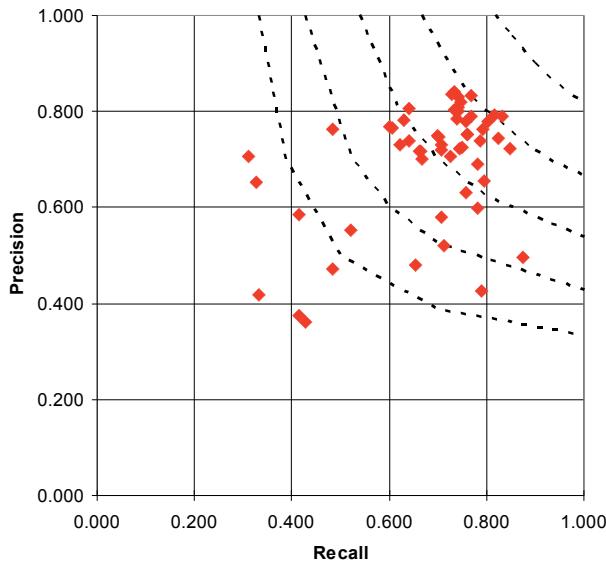


Figure 1: Precision vs. Recall Scatter Plot with F-measure Isobars for GN Macro-averaged Results

3 Methods

3.1 Data Preparation

We handled the data preparation for this task following many of the same procedures developed for the first BioCreative Gene Normalization task [2]. There were, however, some differences, described in greater detail in [3]. We used the GOA annotated records as the basis for selecting documents rich in human genes and proteins. However, the GOA annotators annotate from full text, and we were using only abstracts; furthermore, the GOA annotation process does not include every human gene mentioned in an article, but only specific genes of interest. Finally, we wished to provide a richer linguistic context for the data set, so for each gene, the annotators were asked to flag one string in the text that represented the mention of that gene. This had the effect of supplying a short “evidence passage” for the mention of each gene identifier annotated in the abstract.

To produce the training and test sets, an expert annotator produced a detailed manual annotation of abstracts; the annotator also flagged any annotations about which he had a question. These were checked by the first author. We also performed a small interannotator agreement study, using an additional expert annotator. The results showed ~90% pairwise interannotator agreement. The final training set consisted of 281 abstracts; the blind test set consisted of 262 abstracts.

In addition to the carefully annotated data, we also provided 5000 abstracts from the GOA annotated data; these were sparsely annotated but were likely to contain at least those gene/proteins curated as part of the GOA annotation process.

3.2 Lexical Resources

In addition to the annotated abstracts and the noisy training data, participants were also provided with a lexicon. To create the lexicon, we took the gene symbol and gene name information for each human

EntrezGene identifier from the *gene_info* file from NCBI ([//ftp.ncbi.nlm.nih.gov/gene/DATA](http://ftp.ncbi.nlm.nih.gov/gene/DATA)). We merged this with **name**, **gene** and **synonym** entries taken from UniProt [4]. Suffixes containing "_HUMAN", "1_HUMAN", "H_HUMAN", "protein", "precursor", "antigen" were stripped from the terms and added to the lexicon as separate terms, in addition to the original term. The Hugo Gene Name Consortium (HGNC) **symbol**, **name**, and **alias** entries were also added [5]. We then identified the most often repeated phrases across identifiers as well as those that had numerous matches in the 5000 abstracts of noisy training data. We used these to create a short (381 term) list to remove the most common terms that were unlikely to be gene or protein names but which had entered the lexicon as full synonyms. Examples of entries in this list are "recessive", "neural", "Zeta", "liver", "glycine", and "mediator". This list is available from the BioCreative CVS archive [6]. This left a lexicon of 32,975 distinct EntrezGene identifiers linked to a total of 163,478 unique terms. The majority of identifiers had more than one term attached (average 5.5), although 8,385 had only one.

3.3 Scoring and Revising the Gold Standard

Scoring was done with a python script that matched the gene identifiers returned for each abstract against the gold standard for that abstract. The script also checked for the presence of a textual evidence string for each gene, although this did not effect the actual score, and the textual evidence provided by many submissions did not exactly match the original abstracts. The scoring software was provided to the participants, along with the answer key for the training data.

In order to improve the answer key for the gold standard test set, we did answer pooling to verify the results. The submissions were scored using the preliminary answer key (original gold standard) and then we selected the results of the top ranking (micro-averaged) submission from each team. We pooled the results and re-examined any annotation which disagreed with the gold standard by over 50% of the groups. This led us to reexamine 219 annotations in 126 abstracts. As a result, we added 32 annotations and removed 21 annotations. For the final gold standard, there were a total of 785 gene identifiers for the 262 abstracts.

4 Discussion

Each participating team was required to submit a system description in order to receive their scores. In addition, at a later date, each group was asked to write up a short description analyzing their performance. The observations in this section are based on those write-ups, included in the Workshop Proceedings.

4.1 Analysis of systems

The approaches taken to the gene normalization task were quite varied, but overall, they followed the general pattern below:

- 1) Establishment of a lexical resource to map synonyms against gene identifiers;
 - 2) Tokenization and labelling of the words/terms in the text; this could include special handling for prefixes, suffixes, and enumerations or conjunctions;
 - 3) Matching of candidate mentions in the text against the lexical resource for extraction of the candidate gene identifier(s);
 - 4) Post-processing to remove false positives due to various sources of ambiguity and false matches.

Gene mention detection

A number of teams built directly on their BioCreative GM system (teams 4, 6, 104, 109, 110) to handle steps 1 and 2 above. Several other teams used “off-the-shelf” systems such as LingPipe or ABNER for entity recognition, followed by various post-processing steps.

Lexicon

A number of groups focused on developing and tuning their lexicon. This included enrichment through the addition of further synonyms and pattern-based expansion of the lexicon (e.g., adding variants for Greek as well as Roman suffixes). Other approaches included pruning the lexicon by elimination of highly ambiguous terms or terms that generated false positives (e.g., common words of English or biological terms that were not gene names but occurred in similar contexts, such as cell lines). Teams 4, 13, 34, 109 and 113 explored performance results using different lexicon variants. Interestingly, Team 113 reported higher results (in particular, higher precision) using a smaller, carefully edited lexicon.

Tokenization and Pre-processing

Several teams incorporated a special purpose module for handling abbreviations and gene symbols. In some systems, gene symbols were processed via a separate pipeline; in others, any 3-letter expression was checked for an adjacent full form, and then both forms were used in subsequent term matching. Four teams (4, 34, 42, 109) discussed handling of conjoined forms or enumerations, such as *protein kinase C isoforms alpha, epsilon, and zeta*, or *freac-1 to freac-7*. Team 4 noted in their write-up [7] that an estimated 8% of the names in the development data involved some form of conjunction.

Matching

A number of teams focused on the procedure for matching words in text against terms in the lexicon. Techniques included edit distance, Dice coefficient, Jaro-Winkler distance, percent of matching words, and matching against heuristic patterns. One system (107) used trigram matching – each candidate gene mention was reduced to (letter) trigrams, which were matched against a lexicon.

Post-processing

False positives can come about in several ways: several identifiers can “match” the mention; the mention can also be a word/phrase in English (or in biology), not a gene mention; or the gene may refer to a non-human gene. In some systems, the normalization and disambiguation/filtering were combined into a classification step, where a classifier was trained to distinguish valid gene identifiers from spurious ones, as done in BioCreative I [8].

4.2 Analysis of Results

One advantage of running a series of evaluations is to be able to answer the question: is the research community making progress? To answer this, it is useful to compare the results of Gene Normalization for this BioCreative to the results from the first BioCreative. Table 2 shows a set of statistics for the four tasks, three from the first BioCreative, and the top set, for human gene/proteins, from this BioCreative. The statistics on synonym length, number of synonyms per identifier, and number of identifiers per synonym are all computed relative to the lexicon supplied as part of the task. This is somewhat misleading since, as noted above, many systems used either a richer lexicon, or, in some cases, a lexicon pruned of ambiguous terms. Based on our experiences in BioCreative I, we would have expected human gene/protein normalization to be comparable to Mouse, although we expected that the greater ambiguity in human gene names might cause significant degradation of results. Indeed, our in-house experiments led us to believe that human gene/protein identification might be considerably more difficult. However, the final results are quite comparable, probably due in part to the greater sophistication of this next generation of systems.

	No of Unique IDs	Ave Synonym Length in Words	Ave # Synonyms per Identifier	Ave # Identifiers per Synonym	BioCreative Max Recall @ Precision	BioCreative Max F-measure
Human	32,975	2.17	5.55	(Ambiguity)	0.88 @ 0.50	0.81
Mouse	52,494	2.77	2.48	1.02	0.90 @ 0.43	0.79
Yeast	7,928	1.00	1.86	1.01	0.96 @ 0.65	0.92
Fly	27,749	1.47	2.94	1.09	0.84 @ 0.73	0.82

Table 2: Statistics comparing BioCreative II GN task (human) to BioCreative I tasks (mouse, fly, yeast). Statistics on synonyms are based on lexical resources provided to the participants.

One interesting statistic is the maximum recall reported among the systems. There seem to be several types of difficult cases that have the potential to create a “recall ceiling”:

- Conjoined expressions and range expressions, e.g., *frec-1* to *frec-7*;
 - Short highly ambiguous symbols that can stand for gene families, e.g., AMP.
 - Long names that are descriptions of the gene or paraphrases or permutations of the gene name found in the lexicon, e.g., *alpha1A voltage-dependent calcium channel*

4.4 Improvement Through Voting

We did a simple experiment to determine whether the pooled system responses could perform better than the individual systems. We collected the list of false positive and false negative responses, together with the number of systems that “voted” for each response, based on results from the best (micro-averaged) system run for each team. Using these data, it is straightforward to look at the trade-offs in an (unweighted) voting scheme. Table 3 below shows the pooled system performance at various voting thresholds, in terms of true positives, false positives and false negatives. The top line shows the number of votes needed to record a response. In the range of 6 to 10 out 20 votes, the F-measure for the pooled system is over 0.83 – which is two percentage points higher than the best single team score (F-measure of 0.811). It seems likely that this result could be improved by the use of a more sophisticated weighting scheme.

Min Votes	1/20	2/20	3/20	4/20	5/20	6/20	7/20	8/20	9/20	10/20	11/20	12/20
TP	760	744	737	715	704	692	672	655	633	610	588	565
FP	2534	766	482	331	246	189	155	123	101	75	62	50
FN	25	41	48	70	81	93	113	130	152	175	197	220
R	0.968	0.948	0.939	0.911	0.897	0.882	0.856	0.834	0.806	0.777	0.749	0.720
P	0.231	0.493	0.605	0.684	0.741	0.785	0.813	0.842	0.862	0.891	0.905	0.919
F	0.373	0.648	0.736	0.781	0.812	0.831	0.834	0.838	0.833	0.830	0.820	0.807

Table 3: Summary of True Positives (TP), False Positives (FP), False Negatives (FN) and overall recall (R), precision (P) and F-measure (F) obtained by a simple unweighted voting approach across the best-system pooled data; F-measures above 0.83 are in bold.

Conclusions

Performance on the BioCreative II Gene Normalization task has demonstrated the progress made in this area since the first BioCreative workshop in 2004. This year's assessment involved 20 groups, compared to 8

groups for BioCreative I. The results obtained for human gene/protein identification are comparable to results obtained earlier for mouse and fly, although human gene nomenclature is more complex; three systems achieved an F-measure of 0.80 or above. In addition, a simple unweighted voting algorithm based on the pooled results from all systems achieved over 0.83 F-measure, and it is possible that a more sophisticated voting algorithm could obtain still higher results.

What does this mean in terms of practical performance? The current formulation of the GN task is still quite artificial. A more realistic task would be to provide the capability needed for the protein-protein interaction task: the ability to extract and normalize protein names across multiple species, in full text articles. An important follow-on activity for the BioCreative organizers will be to create a new GN corpus using the texts from the PPI task. This would support a more fine-grained analysis of the PPI results, and would also make it possible to calibrate the current BioCreative tasks against a more realistic (and harder) challenge.

Criteria for a successful evaluation include participation, progress, diversity of approaches, exchange of scientific information, and emergence of standards. We can see all of these happening in the BioCreative evaluation. There is enthusiastic participation in the entire range of BioCreative tasks; the research community is making significant progress as shown by the larger number of high performing systems. There are more groups engaged, and more teams are emerging that combine skills from multiple disciplines, including biology, bioinformatics, machine learning, natural language understanding and information retrieval. There is a healthy variety of approaches being tried. We are seeing exploration of ideas developed in the first BioCreative, such as use of a high recall gene mention “nomination” process, following by a filtering stage. And while the GN task was designed to leverage existing standards, such as EntrezGene identifiers, we are seeing the emergence of reusable component-ware – and a number of high performing systems that are taking advantage of this. As we go forward, the BioCreative Workshop will provide an opportunity to exchange insights and to define the next set of challenges for this community to tackle.

Acknowledgements

This work was supported under National Science Foundation Grant II-0640153. The expert annotation of the data sets was performed by Jeff Colombe; Marc Colosimo provided the second set of annotations for the interannotator agreement study.

References

- [1] Hirschman, L., et al., *Overview of BioCreAtIvE task 1B: Normalized Gene Lists*. BMC Bioinformatics, 2005. **6 (Suppl 1)**: S11.
- [2] Colosimo, M., Morgan, A., Yeh, A., Colombe, J., Hirschman, L., *Data Preparation and Interannotator Agreement: BioCreAtIvE Task 1B*. BMC BioInformatics, 2005. **6 (Suppl 1)**: S12.
- [3] Morgan, A.A., et al. *Evaluating the Automatic Mapping of Human Gene and Protein Mentions to Unique Identifiers*. in *Pacific Symposium for Biocomputing*. 2007. Maui.
- [4] Wu, C.H., et al., *The Universal Protein Resource (UniProt): an expanding universe of protein information*. Nucleic Acids Res, 2006. **34**(Database issue): p. D187-91.
- [5] Wain, H.M., et al., *Genew: the Human Gene Nomenclature Database, 2004 updates*. Nucleic Acids Res, 2004. **32**(Database issue): p. D255-7.
- [6] <http://biocreative.sourceforge.net/>, *BioCreative2 Homepage*.
- [7] Baumgartner, W., et al. *An integrated approach to concept recognition in biomedical text*. in *Second BioCreative Workshop*. 2007. Madrid.
- [8] Crim, J., R. McDonald, and F. Pereira, *Automatically annotating documents with normalized gene lists*. BMC Bioinformatics, 2005. **6(Suppl 1):S13**.

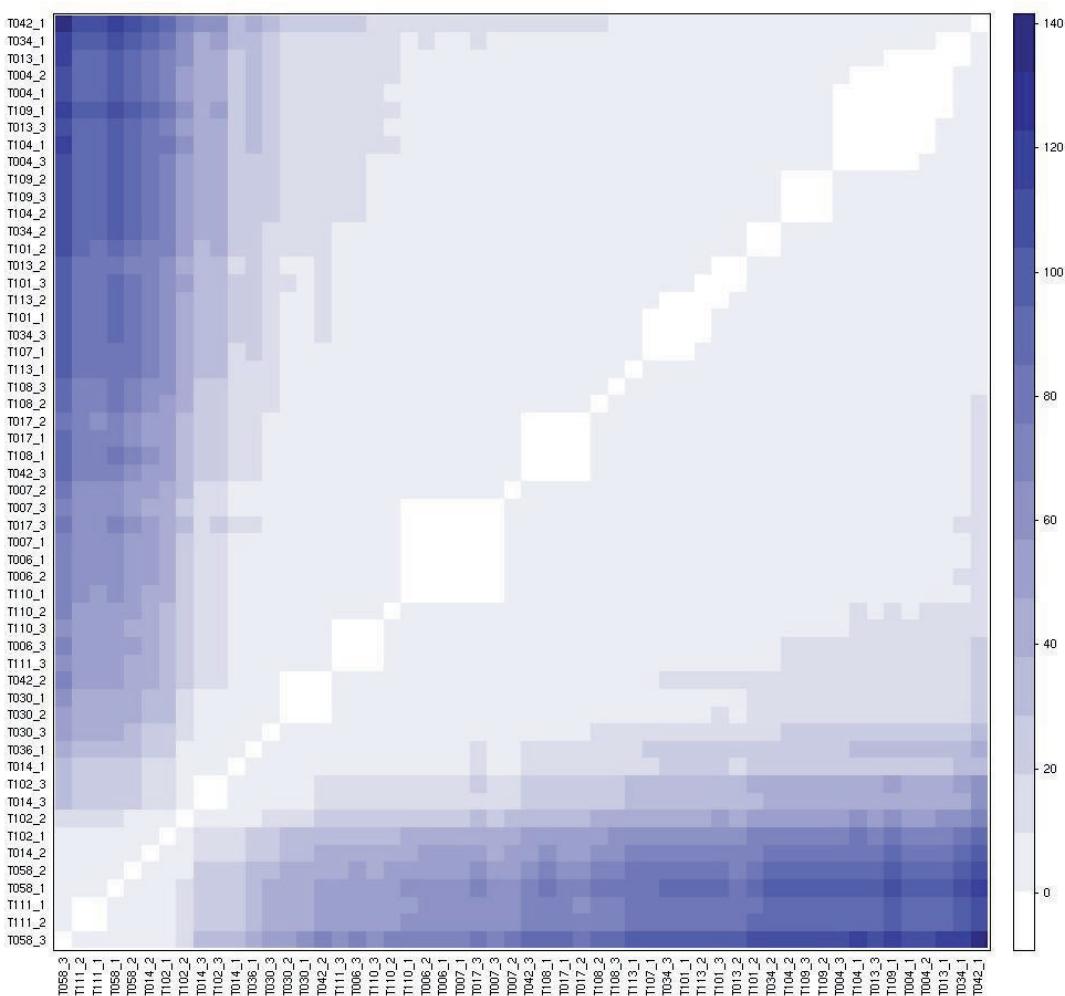
Appendix A: Scores from Gene Normalization Runs

Team_Run	Micro-Average						Macro-Average		
	Recall	Precision	F-measure	True Positive	False Positive	False Negative	Recall	Precision	F-measure
Maximum	0.875	0.841	0.810	687	840	541	0.876	0.898	0.811
Top Quartile	0.767	0.782	0.770	602	271	282	0.817	0.807	0.754
Median	0.750	0.797	0.773	582	149	227	0.832	0.784	0.759
3rd Quartile	0.732	0.735	0.717	575	198	211	0.775	0.773	0.700
Minimum	0.311	0.361	0.370	244	102	98	0.474	0.316	0.342
T004_1	0.734	0.841	0.784	576	109	209	0.876	0.776	0.775
T004_2	0.743	0.829	0.784	583	120	202	0.865	0.784	0.777
T004_3	0.748	0.820	0.782	587	129	198	0.854	0.787	0.772
T006_1	0.601	0.767	0.674	472	143	313	0.794	0.700	0.686
T006_2	0.606	0.767	0.677	476	145	309	0.787	0.705	0.685
T006_3	0.782	0.597	0.677	614	414	171	0.629	0.808	0.666
T007_1	0.707	0.731	0.719	555	204	230	0.755	0.728	0.687
T007_2	0.703	0.746	0.724	552	188	233	0.770	0.725	0.694
T007_3	0.699	0.749	0.723	549	184	236	0.770	0.717	0.688
T013_1	0.768	0.833	0.799	603	121	182	0.848	0.803	0.779
T013_2	0.730	0.835	0.779	573	113	212	0.856	0.770	0.749
T013_3	0.803	0.779	0.790	630	179	155	0.816	0.829	0.773
T014_1	0.485	0.762	0.593	381	119	404	0.783	0.575	0.584
T014_2	0.483	0.471	0.477	379	425	406	0.474	0.517	0.419
T014_3	0.655	0.479	0.553	514	559	271	0.543	0.701	0.555
T017_1	0.708	0.720	0.714	556	216	229	0.764	0.754	0.709
T017_2	0.641	0.806	0.714	503	121	282	0.845	0.701	0.710
T017_3	0.757	0.631	0.688	594	348	191	0.671	0.804	0.688
T030_1	0.661	0.716	0.687	519	206	266	0.736	0.695	0.649
T030_2	0.666	0.702	0.684	523	222	262	0.729	0.698	0.645
T030_3	0.707	0.580	0.637	555	402	230	0.616	0.737	0.617
T034_1	0.815	0.792	0.804	640	168	145	0.815	0.841	0.782
T034_2	0.847	0.723	0.780	665	255	120	0.736	0.870	0.758
T034_3	0.789	0.739	0.763	619	219	166	0.754	0.821	0.740
T036_1	0.713	0.520	0.602	560	516	225	0.562	0.764	0.595

Team_Run	Micro-Average						Macro-Average		
	Recall	Precision	F-measure	True	False	False	Recall	Precision	F-measure
				Positive	Positive	Negative			
T042_1	0.833	0.789	0.810	654	175	131	0.836	0.866	0.811
T042_2	0.875	0.496	0.633	687	699	98	0.567	0.898	0.649
T042_3	0.725	0.707	0.716	569	236	216	0.732	0.760	0.706
T058_1	0.429	0.361	0.392	337	596	448	0.570	0.476	0.382
T058_2	0.415	0.375	0.394	326	543	459	0.611	0.475	0.398
T058_3	0.331	0.419	0.370	260	361	525	0.671	0.371	0.342
T101_1	0.762	0.751	0.756	598	198	187	0.771	0.808	0.741
T101_2	0.743	0.801	0.771	583	145	202	0.820	0.789	0.755
T101_3	0.734	0.804	0.767	576	140	209	0.820	0.779	0.749
T102_1	0.415	0.585	0.486	326	231	459	0.660	0.420	0.431
T102_2	0.521	0.552	0.536	409	332	376	0.619	0.535	0.494
T102_3	0.790	0.425	0.552	620	840	165	0.483	0.814	0.559
T104_1	0.743	0.807	0.774	583	139	202	0.840	0.785	0.773
T104_2	0.758	0.779	0.768	595	169	190	0.804	0.803	0.763
T107_1	0.740	0.784	0.761	581	160	204	0.818	0.776	0.739
T108_1	0.796	0.655	0.719	625	329	160	0.685	0.826	0.708
T108_2	0.782	0.690	0.733	614	276	171	0.723	0.814	0.720
T108_3	0.749	0.726	0.737	588	222	197	0.761	0.785	0.724
T109_1	0.824	0.743	0.781	647	224	138	0.780	0.848	0.775
T109_2	0.792	0.764	0.778	622	192	163	0.806	0.815	0.767
T109_3	0.769	0.790	0.779	604	161	181	0.817	0.806	0.764
T110_1	0.629	0.783	0.698	494	137	291	0.830	0.691	0.685
T110_2	0.641	0.738	0.686	503	179	282	0.794	0.708	0.674
T110_3	0.622	0.732	0.672	488	179	297	0.785	0.698	0.669
T111_1	0.327	0.652	0.436	257	137	528	0.790	0.331	0.362
T111_2	0.311	0.705	0.431	244	102	541	0.828	0.316	0.357
T111_3	0.664	0.717	0.689	521	206	264	0.731	0.706	0.664
T113_1	0.745	0.723	0.734	585	224	200	0.779	0.795	0.733
T113_2	0.761	0.752	0.756	597	197	188	0.782	0.810	0.745

Appendix B: Significance Scores

The table below shows the negative log scores from a two-sided t-test on the macro-averaged f-measure from all runs. The lighter the area, the lower the significance (systems listed on the left and bottom, color scale on the right).





Evaluating the Detection and Ranking of Protein Interaction relevant Articles: the BioCreative Challenge Interaction Article Sub-task (IAS)

Martin Krallinger¹ Alfonso Valencia¹
mkrallinger@cnio.es valencia@cnio.es

¹ Dep. Struct. Comp. Biology Spanish National Cancer Centre (CNIO), Madrid, Spain

Abstract

To extract biological annotations from the literature it is crucial to detect first the articles which are relevant for further manual curation. Although this aspect is important for subsequent information extraction steps, it has often been neglected by previously published protein-protein interaction (PPI) extraction systems. Thus the aim of the Interaction Article Subtask (IAS) was to evaluate the automatic detection and ranking of articles relevant to extract protein interaction information, according to the curation criteria followed by interaction annotation databases. Participants were provided with a labeled training collection of relevant and non-relevant PubMed abstracts. For the collection of test set articles, participants were asked to classify and rank them whether they are relevant to extract protein interactions. A total of 19 teams submitted 51 runs for the IAS. Many participating strategies adapted traditional supervised learning techniques to address this problem. The top scoring systems reached an f-score of over 0.78, an area under the ROC curve (AUC) of around 0.85 and an accuracy of over 0.75.

Keywords: Protein interactions, text categorization, article detection, AUC, article ranking, biological annotations

1 Introduction

It is still a challenge for researchers, and it is also an implicit task in the context of database curation in the biomedical domain to retrieve information of experimentally characterized protein function and interactions [Krallinger and Valencia2006]. Detecting which articles satisfy their information demand in databases such as PubMed is often cumbersome and time-consuming due to the variety of covered disciplines and the growing amount of stored publications. Also the difficulty to detect in this scenario and rank relevant articles using traditional keyword-based search strategies promoted interest in using alternative, more efficient retrieval strategies. In the case of protein interaction-related literature, where a range of experimental methods are being used to characterize interactions (in addition to the variety of language expressions which are used to refer to protein-interaction information) [Azuaje and Dopazo2005] information retrieval and extraction systems have been developed in the past, mainly focusing on aspects related to the automatic extraction of gene and protein interactions or pathways from the literature [Hoffmann et al.2005]. Most of these techniques are based on co-occurrence analysis of proteins names in combination of interaction language expressions, but lack a detection system for interaction relevant articles, what is probably one of the most relevant aspects for curation. Indeed importance of using a system to detect interaction relevant articles for database curation has been realized through applications such as PreBIND, which was used to improve the curation efficiency of the BIND interaction database [Donaldson et al.2003]. The importance of characterizing protein interactions to understand not only the functional role of individual proteins but also the organization of entire biological processes, in addition to the efforts which have been made to

standardize manually curated protein interactions [Zanzoni et al.2002, Hermjakob et al.2004], make the automatic detection of protein interaction relevant literature a crucial exercise. Some previous attempts to evaluate text categorization in the biomedical domain have been made in the context of the TREC Genomics track [Hersh et al.2005], but did not specifically focus on protein interaction annotation. We thus proposed a specific sub-task in the Second BioCreative challenge which is concerned with the detection of protein-protein interaction relevant articles from PubMed titles and abstracts.

2 IAS Overview

Participants were provided with a collection of PubMed abstracts, allowing them to build and train their interaction literature classification systems during June-October 2006. The articles were previously selected by the IntAct and MINT interaction databases. In case of the interaction-relevant records, they were mainly derived from these two interaction databases, and therefore associated annotations existed for each article identifier.

In general, one can consider mainly three curation strategies generally followed by database annotators in biology (although there might be exceptions and hybrid approaches). In the first approach biologists take the 'whole' PubMed or a large collection of journals as base for detecting annotation relevant articles, often using keyword search strategies to find the relevant articles (thematic curation). In the second approach, curators exploit citations provided by other databases or researchers, to extract additional information from these articles (citation overlap/recommendation-based curation). Finally, in the third strategy, biologists curate and check each/all the articles published by a given journal, usually over a certain period of time (exhaustive curation).

After the training period, participants received a test set collection of unlabeled abstracts and had to classify them automatically whether they are interaction-relevant or not, also ranking them according to relevance and non-relevance respectively.

As output of their systems, the participants were asked to return a ranked list of article identifiers based on their relevance for protein interaction annotation. We advised the participants that human re-ranking and manual inspection of the predictions were not allowed and that by submitting results, the groups agreed to have their submissions made public in an anonymous form at the end of the evaluation.

Submissions consisted in two separated lists, one for predicted relevant (positive) and one for predicted non-relevant (negative) articles. The reason why a non-relevant article ranking was requested is based on the idea to provide database curators with an exclusion list of articles for the curation process.

Although the curation process is based on full text articles, only PubMed titles and abstracts were provided for this sub-task. This reflects the actual textual data which is freely available, as in practice there are still serious limitations to obtain large collections of updated full text articles for most biomedical journals. It is also a way to explore the limits of abstract-based detection of annotation relevant articles. Finally most of the manual curation strategies start in practice with an initial reading of the article abstracts, followed only by a more careful examination of the full text articles (especially figure legends and experimental characterizations) in cases where the abstract provides enough hints that the article is curation-worthy. This implies that there are some cases where the use of abstracts alone is not enough to fully determine the annotation-relevance.

As MINT and IntAct are doing an exhaustive curation for a specified list of journals, it is of particular interest to filter articles based on annotation relevant information, thereby increasing curation efficiency. We therefore analyzed how good the participants strategies were in both detection and ranking of relevant articles, as well as for articles which do not contain interaction relevant information.

3 Data collections

In order to get access to the data collections used for the protein interaction task, participants had to register, providing their corresponding team member and contact information. Each team then received a training collection to develop their systems from June to October 2006. After a period of several months a smaller test collection was released, for which predictions had to be returned within less than two weeks. The submissions had to be made in a predefined format together with a short system description.

3.1 Training data

The construction of a suitable training set for the IAS exploited the content of existing interaction databases, namely IntAct and MINT. The motivations behind this data selection strategy were the following ones:

- Explore the usability of existing citation collections derived from biological annotation databases for the detection of curation-relevant articles.
- Pinpoint the main challenges for selecting and retrieving suitable article collections, based on existing database citations.
- Evaluate the use of abstract-based article classification and ranking versus manually curated articles.

The annotation records of both interaction databases are freely accessible for download and share a common annotation standard based on the HUPO PSI Molecular Interaction Format. For a numeric overview on the size of the provided training and test set, please refer to table 1. The training collections where distributed using a simple XML-like format.

Three abstract collections were included in the training set for this subtask:

1. The *Positives (P) collection* (i.e. physical protein-protein interaction relevant articles) was based on a set of PubMed articles which are relevant for protein interaction curation in the sense of the annotation process and guidelines used by the MINT and IntAct databases. This means that the corresponding full text articles have been used to extract manual annotations and therefore meet the underlying curation standards used to extract experimentally verified protein interaction information. The initial collection contained articles resulting from exhaustive curation as well as from thematic curation. Some articles were removed from this collection mainly because either no corresponding abstracts could be retrieved from PubMed or they corresponded to results obtained by large scale experiments. As the actual curation was done based on the full text articles as opposed to the abstracts, it is conceivable that in some cases the abstracts in this collection may lack sufficient information to be considered as interaction-relevant. The initial positive training collection consisted of 3,536 PubMed titles and abstracts distributed together with the corresponding PMID and the article source (journal and publication date).
2. The *Negatives (N) collection* (i.e. non-relevant articles) consisted exclusively in journal titles and abstracts rejected during exhaustive curation. These articles have no associated annotation records extracted by the domain expert curators and are thus not relevant for protein interaction annotation. I. e. only for those journals, for which exhaustive curation had been carried out, negative training instances were available. The provided negative collection contained a total of 1,959 entries. The training collections of the positive and negative instances were not balanced; participating systems had to address the resulting class imbalance.

3. Finally, we also included a collection of *likely Positive* (P_L) articles, consisting of PubMed citations which had been extracted from protein interaction annotations curated by other interaction databases (including BIND, HPRD, MPACT and GRID). This additional large collection constitutes a *noisy* data set in the sense that the corresponding databases have different annotation standards compared to MINT and IntAct (for instance regarding the curation of genetic interactions) and thus have not been included as part of the ordinary positive training collection. This collection consisted of a total of 18,930 records.

No restrictions in terms of using additional resources or data collections for the purpose of system development and training were imposed on the participating teams. Therefore, also additional resources such as resulting from gene mention detection or associated MeSH terms could be exploited, as is also done in real life situations.

3.2 Test data collection

In order to perform a comparative assessment of the various participating systems, a common test data collection was provided to all the participants. This data set consisted in a collection of PubMed records (article titles and abstracts) in a format compliant with the training collection, but without providing the corresponding article source information as well as without the actual class label (relevant or not relevant). Most of the articles in the test set resulted from exhaustive curation of recent publications from specified journals (such as the EMBO Journal or FEBS letters) published over a predefined period of time. The resulting annotations from the curation of these articles were held back by the interaction databases until the competition was over. Some of the initial test set articles supplied by the database curators had to be removed from the test set, because no PubMed abstract was available. An additional criteria for the construction of the test set was to make sure that neither publication date nor journal name could be used as a relevant discriminative feature for classifying the articles. The relevant and non-relevant entries were randomly shuffled so that the article order in the test collection could also not be used to differentiate relevant from non-relevant records. The resulting test set collection of 750 entries was an actual subset from the initial collection provided by the database curators. One of the databases also provided a small number of *un-curatable* abstracts, meaning that the associated full text articles were not worthwhile to curate (to complicated and from a very specific scientific sub-discipline) or the abstract was misleading, meaning that protein interactions were mentioned in the abstract but the full text article lacked the experimental characterization for the proposed interactions. These articles were also removed from the test collection.

The resulting initial IAS test set consisted in 375 positive (relevant) and negative (non-relevant) entries respectively. Nevertheless, during the post-evaluation period, several records were revised and finally removed from the initial test collection. Thus the revised test set contained a slight imbalance, consisting in of 338 interaction relevant articles and 339 non-relevant records (677 total instances).

4 Overview of the used systems

A range of different techniques were applied in order to detect protein interaction relevant articles based on PubMed abstracts. To allow a more straightforward comparison of the participating systems, the registered teams were asked to fill in a basic system description questionnaire when submitting their predictions. Table 1 shows the normalized output of this questionnaire. Most of the systems (15 out of 19) did not make use of any additional resources, and also did not exploit (in case of their final submissions), the collection of *likely Positive* articles. The paraphrase of the posed questions is as follows:

Q1: Use of additional training data (in addition to the provided one).

Q2: Use of additional noisy training data of TP abstracts.

Q3: Use of machine learning (ML) approaches.

Q4: Used ML techniques.

- Q5:** Use of protein name tagging.
- Q6:** Use of NLP technique components.
- Q7:** Used NLP components.
- Q8:** Use of Bio-NLP components.
- Q9:** Use of external lexical resources.
- Q10:** Processing using sentence units.
- Q11:** Processing using whole abstracts as units.

A common characteristic of the majority of the participating strategies was the usage of machine learning techniques (17 out of 19), Support Vector Machines (SVM), naïve Bayes and Maximum Entropy classifiers being the most frequently used methods. Regarding the natural language processing (NLP) components often integrated into these systems, stemming and Part-of Speech tagging were the most common ones. Surprisingly, only very few systems exploited Bio-NLP applications such as protein name taggers or adapted existing lexical resources such as biological ontologies for detecting interaction relevant articles. Although a considerable number of teams used sentences as their processing unit, most of the teams considered whole abstracts and bag-of word approaches.

5 Results and evaluation

Each registered team was allowed to submit up to three runs. From a total of 19 teams, 51 runs were received. The participants had been asked to provide ranked predictions for all the entries (articles) in the test set to make the different system predictions comparable. Two files in a predefined, tabulator-separated prediction format were received for each run, one corresponding to the predicted interaction-relevant articles (P), the other corresponding to the non-interaction relevant articles (N). For each run, the precision, recall, f-score, accuracy and area under the ROC curve (AUC) have been calculated by comparing the predicted labels to the manually curated test set articles.

$$Precision = \frac{TP}{TP + FP}; \quad Recall = \frac{TP}{TP + FN}; \quad (1)$$

$$f-score = \frac{2 * Precision * Recall}{Precision + Recall}; \quad Accuracy = \frac{TP + TN}{P + N} \quad (2)$$

where TP: number of True Positive predictions, FP: False Positives, FN: False Negatives, TN: True Negatives, P: total number of Positives and N: total number of Negatives. For the actual calculation of the AUC, we used the standard R-package ROCR, which integrates the most common evaluation metrics to assess the performance of classifiers [Sing et al.2005]. The teams received the evaluated results using the initial IAS test set. Here we present the evaluation using the revised test set.

The overall results obtained by each of the teams can be seen in table 2, which provides the obtained scores for each single run, while figure 1 shows the corresponding precision-recall plot. The submission with the highest the highest AUC (0.8554) has been submitted by team 6, with a precision of 0.7080, a recall of 0.8609 and a f-score of 0.7770. The underlying system is characterized by the use of a SVM classifier and careful pre-processing steps as well the integration of a series of traditional NLP strategies, like stemming, POS-tagging, sentence splitting and shallow parsing. This system also exploited domain specific NLP applications for protein name detection and abbreviation resolution. The highest f-score was obtained by team 57 (0.7800), which also used a SVM classifier. All the top scoring teams had in common the usage of a SVM classifier together with word stemming. Previous work related to the classification of interaction describing sentences showed how useful SVM classifiers in combination with word stemming are [Krallinger et al.2006] for related scenarios.

TEAM	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
T4	Y	N	Y	SVM, NB	N	Y	stemming, POS tagging, shallow parsing	N	N	Y	Y	Y
T6	Y	N	Y	SVM	Y	Y	stemming, POS tagging, sentence splitting, tokenisation, shallow parsing, abbreviation resolution	Y	Y	Y	Y	N
T7	Y	N	Y	SVM	N	N	N	N	N	N	Y	N
T11	N	Y	Y	SVM, SVD, PCA, VM, SD, NN	Y	Y	stemming, TFIDF, bigrams	Y	N	N	N	Y
T14	N	N	Y	SMO	Y	N	N	N	Y	N	Y	Y
T19	N	N	N	N	Y	Y	commonwords, PPI connection keywords	N	N	N	Y	Y
T27	N	N	Y	NB	N	Y	stemming	N	N	N	Y	N
T28	N	N	Y	SVM	N	Y	stemming	N	N	N	Y	N
T30	N	Y	Y	NB	Y	Y	stemming	N	Y	Y	N	Y
T31	N	N	Y	Unsupervised iterative pattern learning	Y	N	N	N	Y	Y	N	Y
T37	N	N	Y	NB	N	N	N	N	N	N	Y	N
T41	N	Y	N	N	N	Y	stemming, POS tagging, parsing, chunking	N	Y	Y	Y	Y
T44	N	N	Y	SVM, CBR	N	Y	stemming	N	N	N	N	N
T48	N	N	Y	ME, NB, VP learner	N	N	N	N	N	N	Y	N
T49	N	N	Y	NB, ME	Y	Y	lemmatizer, sentence splitting, stop list	Y	N	Y	Y	Y
T51	N	N	Y	ME	Y	Y	stemming, NER	Y	N	Y	Y	N
T52	Y	N	Y	SVM	Y	N	N	N	N	N	Y	N
T57	N	N	Y	SVM	N	Y	stemming	N	N	N	Y	N
T58	N	Y	Y	NB, ME, SVM	N	N	N	N	N	N	Y	N
ALL	Y4, N15	Y4, N15	Y17, N2	SVM:9, NB:5, ME:4	Y9, N10	Y12, N7	stemming:10, POS tagging:3, sentence splitting:2, shallow parsing:2	Y4, N15	Y5, N14	Y7, N12	Y16, N3	Y7, N12

Table 1: IAS-questionnaire

This table shows an overview of the characteristics of the participating systems based on the IAS-questionnaire. Y: Yes, N: No; SVM: Support Vector Machines, NB: Naïive Bayes, ME: Maximum Entropy, SVD: Singular Value Decomposition, PCA: Principal Components Analysis, VM: Vector Model, SD: Spam detection, NN: Nearest Neighbors, CBR: Case Based Reasoning, NER: Named Entity Recognition.

Q1: Use of additional training data (in addition to the provided one). **Q2:** Use of additional noisy training data of TP abstracts. **Q3:** Use of machine learning (ML) approaches. **Q4:** Used ML techniques. **Q5:** Use of protein name tagging. **Q6:** Use of NLP technique components. **Q7:** Used NLP components. **Q8:** Use of Bio-NLP components. **Q9:** Use of external lexical resources. **Q10:** Processing using sentence units. **Q11:** Processing using whole abstracts as units.

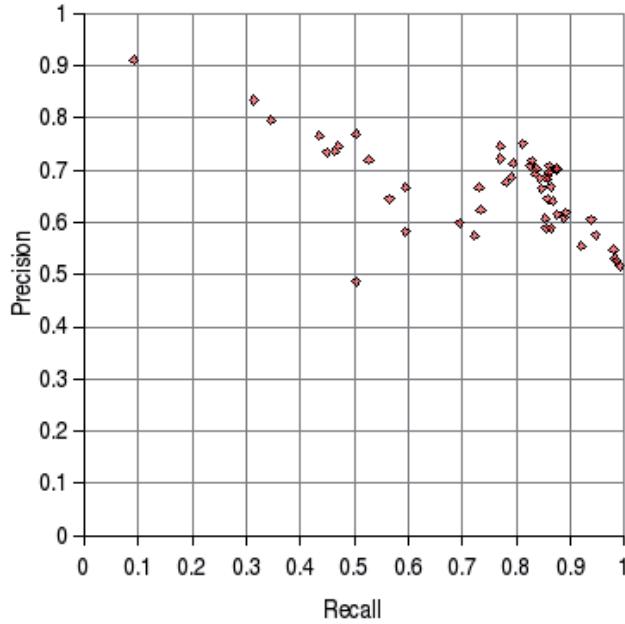


Figure 1: Recall-precision plot of the received runs. This plot shows the recall vs precision of all the submitted runs. Most of the systems focused on obtaining a high recall, with a modest precision. The performance of most systems ranged was characterized by a recall between 0.7 and 0.9 and a precision between 0.6 and 0.8.

Team	Run	Precision	Recall	F-score	AUC	Accuracy
T4	1	0.7040	0.8373	0.7649	0.7495	0.7430
T4	2	0.6061	0.9379	0.7364	0.5529	0.6647
T4	3	0.7128	0.7929	0.7507	0.7479	0.7371
T6	1	0.7080	0.8609	0.7770	0.8554	0.7533
T7	1	0.6851	0.8432	0.7560	0.8270	0.7282
T7	2	0.6682	0.8639	0.7535	0.7875	0.7179
T7	3	0.6840	0.8580	0.7612	0.8318	0.7312
T11	1	0.6411	0.8669	0.7371	0.7995	0.6913
T11	2	0.7222	0.7692	0.7450	0.7567	0.7371
T11	3	0.6769	0.7811	0.7253	0.7013	0.7046
T14	1	0.7343	0.4497	0.5578	0.7500	0.6440
T14	2	0.7371	0.4645	0.5699	0.7561	0.6499
T14	3	0.7465	0.4704	0.5771	0.7570	0.6558
T19	1	0.6247	0.7337	0.6748	0.6765	0.6470
T19	2	0.6453	0.5651	0.6025	0.6765	0.6278
T27	1	0.5886	0.8550	0.6972	0.6812	0.6292
T27	2	0.5554	0.9201	0.6927	0.6244	0.5923
T27	3	0.6076	0.8521	0.7094	0.6945	0.6514
T28	1	0.7507	0.8107	0.7795	0.8471	0.7710
T28	2	0.7471	0.7692	0.7580	0.8150	0.7548

Team	Run	Precision	Recall	F-score	AUC	Accuracy
T28	3	0.6864	0.7899	0.7345	0.7993	0.7149
T30	1	0.5826	0.5947	0.5886	0.6197	0.5849
T30	2	0.4871	0.5030	0.4949	0.5643	0.4874
T30	3	0.5995	0.6953	0.6438	0.6581	0.6160
T31	1	0.6678	0.5947	0.6291	0.6714	0.6499
T31	2	0.7206	0.5266	0.6085	0.6793	0.6617
T31	3	0.7959	0.3462	0.4825	0.6793	0.6292
T37	1	0.5480	0.9793	0.7028	0.6976	0.5864
T37	2	0.5755	0.9467	0.7159	0.7468	0.6248
T37	3	0.5312	0.9822	0.6895	0.6550	0.5583
T41	1	0.6098	0.8876	0.7229	0.7535	0.6603
T41	2	0.6154	0.8757	0.7228	0.7720	0.6647
T41	3	0.6193	0.8905	0.7306	0.7714	0.6721
T44	1	0.6888	0.8580	0.7642	0.7320	0.7356
T44	2	0.6459	0.8580	0.7370	0.5970	0.6942
T44	3	0.7081	0.8254	0.7623	0.7433	0.7430
T48	1	0.9118	0.0917	0.1667	0.6572	0.5421
T48	2	0.5887	0.8639	0.7002	0.6422	0.6307
T48	3	0.8346	0.3136	0.4559	0.6904	0.6263
T49	1	0.5261	0.9852	0.6859	0.7968	0.5495
T49	2	0.5170	0.9911	0.6795	0.7990	0.5332
T49	3	0.5741	0.7219	0.6396	0.5894	0.5938
T51	1	0.7179	0.8284	0.7692	0.8412	0.7518
T52	1	0.6929	0.8343	0.7570	0.8057	0.7326
T52	2	0.6651	0.8462	0.7448	0.8146	0.7105
T57	1	0.7031	0.8757	0.7800	0.8194	0.7533
T57	2	0.7024	0.8728	0.7784	0.8151	0.7518
T57	3	0.6962	0.8609	0.7698	0.8054	0.7430
T58	1	0.7656	0.4349	0.5547	0.7326	0.6514
T58	2	0.7692	0.5030	0.6082	0.7578	0.6765
T58	3	0.6676	0.7308	0.6977	0.7554	0.6839

Table 2: IAS-result

Most of the received runs had a consistently higher recall when compared to the precision. Although the f-score provides a balanced view of both, it is often not the ideal metric for evaluating the usefulness of applications. In real-life scenarios, the most relevant evaluation metric is tightly dependent on the underlying end user demands and the available amount of data. For instance in case of exhaustive curation, a high recall might be more desirable, while in case of thematic curation against the whole literature database, high precision and efficient relevance ranking might have a greater practical value. In this respect, team 48 obtained the highest precision (0.9118) and team 49 the highest recall (0.9911). As can be seen from table 1, only two teams did not make use of machine learning techniques. Although they did not rank among the top scoring participants, the used strategies might be interesting for situations where training data is not available or too noisy.

When comparing the average scores for all the evaluated runs with the majority voting results, it is clear that the agreement of different runs on the classification resulted in improved results. The average precision over all the runs is of 0.6659 while the result for the majority voting is of 0.7078. The same is true for the recall (0.7492 and 0.8817), f-score (0.6793 and 0.7852) as well as accuracy (0.6676

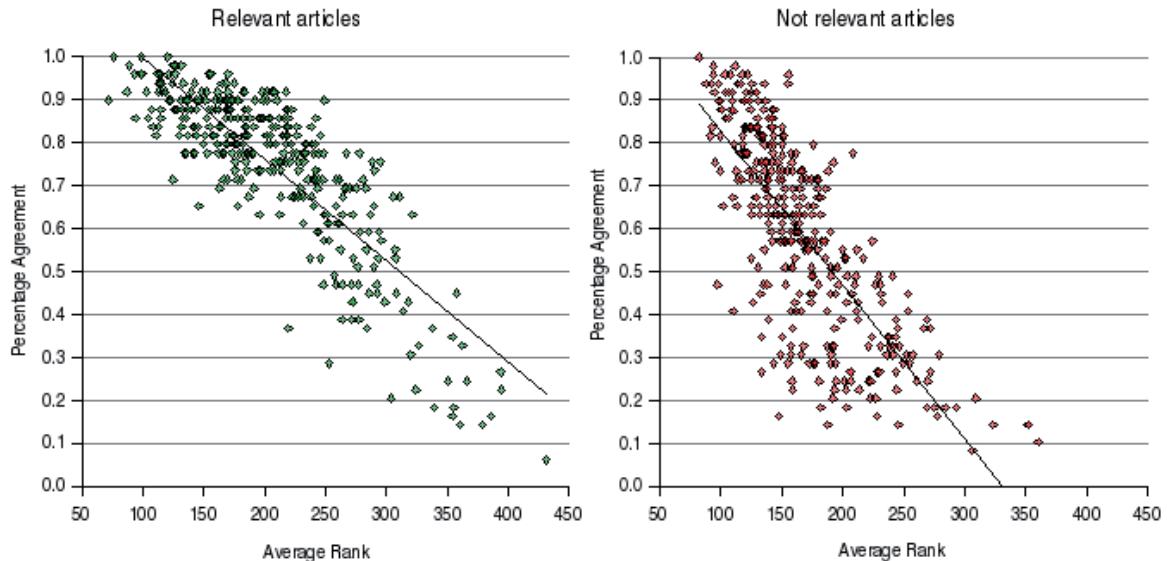


Figure 2: Agreement and average rank. This figure shows the relationship between the prediction agreement of the different runs with respect to the average rank for the relevant and the non-relevant articles. The overall agreement was lower in case of the non-relevant articles between the different systems ($R^2_{relevant} = 0.7$ vs. $R^2_{non-relevant} = 0.59$). Also, in general, the higher the average rank of the article, the more systems agreed on the correct classification.

and 0.7592). When taking into account the relationship between the average rank of the articles with respect to the agreement between the different runs in terms of the correct classification, it seems that in general, the higher the agreement between the different systems on the correct class, the higher is also the average rank of this article (see figure 2).

6 Discussion and Conclusions

The results of the IAS task are promising and show that in general the detection of protein-interaction relevant articles from PubMed titles and abstracts is feasible to certain extent. A comparison with systems using the corresponding full text articles is currently missing, but would certainly show better the boundaries of abstract-based interaction article classification. Similar systems could in principle be adapted to assist biologists in certain steps within the curation process for other biological annotation types, such as gene regulation or cellular localization of proteins.

A deeper analysis of the evaluated results showed some of inherent challenges when using abstracts alone. In case of the articles with high percentage of true positive predictions, the titles and abstracts were in general characterized by a high density of not only words or expressions related to protein interactions such as 'interacts', 'binding', 'interacting partner' or 'interaction of', but also mentioned the actual names of the methods used to characterize experimentally these interactions. In case of the test set article with PMID 16828757 expressions such as 'yeast two-hybrid screen', 'co-immunoprecipitation' and 'in vitro binding assays' can be found.

Many of the false negative articles corresponded to cases where gene regulation or gene expression mechanisms were mentioned. These abstracts are often relevant to both protein interactions as well as

genetic interactions. For example the article with PMID 16547462 describes oligomeric transcription factors. Also cases where domain-specific expressions were used to describe specific interaction types proved more difficult to classify as relevant by some of the systems. In the test set entry PMID 16321977 a methylation event is described: 'Mtq2p methylates Sup45p', or in PMID 16330551 self-oligomerization of the fibrillar protein alpha synuclein is explained.

As for false positive articles, several general characteristics could be distinguished. Surprisingly, some recurrently mentioned certain well characterized proteins, such as EGF or EGFR (e.g. 16316986). One potential reason for this fact might be that they are often mentioned in the positive training collection. Also a considerable number of FP abstracts describe interaction events between proteins and DNA, making use of words such as 'complex' but without referring to protein-protein complexes. This is the case for PMID 16440001, where expressions such as 'complex binding to DNA' are found. Also, articles which describe interaction events between macromolecular structures or cellular components seem to be more challenging (e.g. the association of telomeres with the nuclear envelope in PMID 16467853).

Some of the used methods were also sensitive with respect to the length of the abstracts predicting rather short abstracts as not relevant and very long abstracts as relevant. This is reflected in the average length of the FN articles resulting from the majority voting system (155 words) compared to the FP abstracts (174 words).

References

- [Azuaje and Dopazo2005] F. Azuaje and J. Dopazo. 2005. *Data Analysis and Visualization in Genomics and Proteomics*. Wiley, West Sussex, England.
- [Donaldson et al.2003] I. Donaldson, J. Martin, B. deBruijn, C. Wolting, V. Lay, B. Tuekam, S. Zhang, B. Baskin, G.D. Bader, K. Michalickova, T. Pawson, and C.W. Hogue. 2003. PreBIND and Textomy—mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics.*, 4:11.
- [Hermjakob et al.2004] H. Hermjakob, L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman, and R. Apweiler. 2004. IntAct: an open source molecular interaction database. *Nucleic Acids Res.*, 32:D452–D455.
- [Hersh et al.2005] W. Hersh, A. Cohen, J. Yang, R.T. Bhupatiraju, P. Roberts, and M. Hearst. 2005. Trec 2005 genomics track overview. *TREC Notebook*.
- [Hoffmann et al.2005] R. Hoffmann, M. Krallinger, E. Andres, J. Tamames, C. Blaschke, and A. Valencia. 2005. Text mining for metabolic pathways, signaling cascades, and protein networks. *Sci STKE*, 283:pe21.
- [Krallinger and Valencia2006] M. Krallinger and A. Valencia. 2006. Text-mining and information-retrieval services for molecular biology. *Genome Biol.*, 6:224.
- [Krallinger et al.2006] M. Krallinger, R. Malik, and A. Valencia. 2006. Text Mining and Protein Annotations: the Construction and Use of Protein Description Sentences . *Genome Inform Ser Workshop Genome Inform*, 17:121–130.
- [Sing et al.2005] T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer. 2005. ROCR: visualizing classifier performance in R. *Bioinformatics*, 21:3940–3941.

[Zanzoni et al.2002] L. Zanzoni, A. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni. 2002. MINT: a Molecular INTeraction database. *FEBS Lett*, 513:135–140.



Assessment of the Second BioCreative PPI task: Automatic Extraction of Protein-Protein Interactions.

Martin Krallinger¹ Florian Leitner¹ Alfonso Valencia¹
mkrallinger@cnio.es fleitner@cnio.es valencia@cnio.es

¹ Dep. Struct. Comp. Biology Spanish National Cancer Centre (CNIO), Madrid, Spain

Abstract

There are several important aspects when providing useful tools to assist biologists in extracting biological annotations from the literature. A crucial point is the correct identification and association of mentioned interactor proteins to their corresponding database entries (e.g. SwissProt record IDs). Not only the individual interactors, but also the correct binary interaction pair needs to be extracted. Biological annotations of protein interactions are associated to qualitative information with regard to the interaction detection experiments which have been carried out to characterize the given interaction. Finally, textual passages which summarize the mentioned interaction are relevant for efficient curation and for human interpretation. All these aspects have been addressed in the Protein-Protein Interaction (PPI) task, in the form of several sub-tasks, each focusing on one of the above-mentioned points, namely the Interaction Pair Sub-task (IPS), the Interaction Method Sub-task (IMS) and the Interaction Sentence Sub-Task (ISS). Teams which extracted normalized protein interaction pairs from full text articles reached an f-score of 0.3. The highest precision obtained for the IPS was of 0.39. When considering the detection of the normalized individual interactor proteins, the highest f-score was of 0.48 with a precision of 0.56. In case of the correct association of full text articles to an ontology of controlled vocabulary terms for interaction detection methods (MI-ontology), the best participant achieved a precision of 0.67. As for the retrieval of the best interaction-summarizing passages, 19% of the passages submitted by one of the teams could be mapped to the previously manually extracted best interaction-describing text passages.

The PPI task covers all the relevant steps for the extraction of protein interaction annotations from full text articles. It shows the main potentials as well as difficulties encountered by participating text mining systems in extracting biological annotations when compared to manual human curation. In particular, the performance of the participating strategies was affected by the protein interactor normalization (without any restriction of the associated organism source), the retrieval of interaction text descriptions which span multiple sentences, as well as by implicit difficulties when processing full text articles.

Keywords: Protein-protein interactions, biological annotations, passage retrieval, protein normalization, interactor, binary interaction, controlled vocabulary, interaction detection method

1 Introduction

The study of protein interactions is one of the most pressing biological problems. Characterizing protein interaction partners is crucial to understanding not only the functional role of individual proteins, but also the organization of entire biological processes.

The development of high throughput experimental technologies, such as yeast two-hybrid screening [Uetz et al.2000] or affinity purification coupled with mass spectroscopy, is now making it possible to study protein interactions on a much larger scale by means of bioinformatics approaches [Valencia and Pazos2002]. One limitation of these large-scale experiments is their accuracy. Protein interaction databases have been developed [Hermjakob et al.2004, Zanzoni et al.2002] to integrate protein interaction information from these disparate sources, e.g., high throughput methods as well as

carefully experimentally characterized individual protein interactions. Databases such as IntAct and MINT provide interaction information in the form of well structured database records in standard formats, constituting a useful resource for both biologists as well as bioinformaticians.

Because the molecular biology literature provides detailed descriptions of protein interaction experiments specifying the individual interaction partners, as well as the corresponding interaction types, it has been exploited as a resource to derive protein interaction records for interaction databases. Due to the rapid growth of the biomedical literature and the increasing number of newly discovered proteins, it is becoming difficult for the interaction database curators to keep up with the literature by manually detecting and curating protein interaction information.

This promoted the implementation of information extraction and text mining techniques to automatically extract protein interaction information from free texts. A number of approaches have been published, (see [Ono et al.1999, Daraselia et al.2004, Blaschke and Valencia2002] for some of the strategies). Nevertheless, a large-scale evaluation of different methods applied to existing protein interaction databases is still missing. To produce high quality training and test data collections, as well as to set up community-wide experiments which can result in relevant and useful systems, the collaboration with experts in protein interaction databases is crucial.

2 PPI Task motivation

One of the main limitations for the development and evaluation of protein-protein interaction extraction methods from text is the lack of Gold Standard training data sets. This makes it cumbersome to compare existing automated extraction methods, as most results are reported using author-specific evaluation data sets; furthermore, some systems have only been evaluated using article abstracts.

In practice, biologists who search for protein interactions are not limited to abstracts, but consider full text articles to derive protein interaction information. Also the type of protein interaction and the experimental method used to determine whether two proteins interact are important information preserved in expert-curated databases.

For BioCreAtIvE II, the protein-protein interaction task focuses on the prediction of protein interactions from full text articles. The second BioCreAtIvE challenge represents a joint effort of expert database curators with experience in protein interaction annotation and experts in evaluating information extraction systems adapted to the biology domain.

Among the main goals posed in this task are:

1. Determine the state of the art in extraction of protein-protein interaction;
2. Produce useful resources for training and testing protein interaction extraction systems;
3. Learn which approaches are successful and practical;
4. Monitor interesting new approaches;
5. Provide the biology community with useful tools to extract protein-protein interactions from texts.

This second BioCreAtIvE challenge provided the opportunity for participating systems to take advantage of the underlying collaboration with domain experts while addressing a real life task. The training and test data sets are characterized by in-depth annotations of protein interactions in full text articles.

3 Sub-task Descriptions

This article describes the evaluation of the three PPI sub-tasks which used full text articles as part of the test and training collections, namely the Interaction Pair Sub-Task, the Interaction Methods Subtask and the Interaction Sentences Sub-task.

3.1 Protein Interaction Pairs Sub-task

Results of experimental interaction characterization studies are often described in peer reviewed literature articles. For domain experts to extract manually such protein interactions from the literature is a time consuming process. Here the aim was to identify pairs of interacting proteins from full text articles. The individual proteins of a given interaction pair should be uniquely identified (normalized) by their corresponding UniProt accession numbers (although UniProt IDs were also allowed).

Together with the test set and the training set, a 'light version' of UniProt (i. e. without sequences, features, etc., only containing fields relevant for the task) was distributed to provide a common reference collection for the normalization of the proteins and to avoid inconsistencies resulting from using different database releases. Only records contained in this release were considered for evaluation of the test set predictions, although in real life a single database is often not sufficient to cover all the proteins mentioned in the literature.

The interaction databases IntAct and MINT curate all interactions which can be classified as interaction types (MI:0190) colocalisations (MI:0403) and physical interactions (MI:0218), as well as all the corresponding child nodes. This means that genetic interactions (gene regulation) are not taken into account for extracting annotations. As system input (training data) the participants received a collection of full text articles with the associated interaction pairs curated from these articles, as well as the corresponding gene mention symbols and synonyms (the 'alias type' node of the Molecular Interaction Ontology, MI:0300). For a more detailed description of the used data sets refer to the data collection section.

As test set a collection of full text articles was provided. The participating teams had to provide, for each article, a ranked list of normalized protein-protein interaction pairs. The evaluation of the submitted predictions was done by calculating the precision and recall of the submitted protein interaction pairs for each article compared to the previously manually extracted ones. The interacting proteins in the training and test sets were not restricted to a single organism source, so in principle for the linking step to the UniProt database entry, inter-species protein name ambiguity had to be taken into account.

3.2 Protein Interaction Method Sub-task (IMS)

In order to obtain reliable protein interaction information, it is necessary that these interactions have been experimentally confirmed. For annotation purposes, as well as to judge the quality of protein interactions, it is important to know exactly which methods have been applied to detect those interactions, as each method has also an implicit degree of reliability. Generally for annotation purposes, most biological curators do not extract annotations which lack experimental support.

In case of protein-protein interaction annotations, considerable effort has been made to develop a controlled vocabulary (CV) for interaction detection methods, the Molecular Interaction (MI) ontology [Orchard et al.2005].

This sub-task was concerned with the identification of the type of experiment which was used to confirm a given protein-protein interaction. The experimental method used to detect the interaction described in the article had to be mapped into the controlled hierarchical vocabulary of experimental methods of the Molecular Interaction (MI) ontology. This implied that participants had to associate the articles to the correct concepts within the interaction detection methods (MI:0001) branch of the MI ontology. The MI ontology also provides additional information for each concept such as their

definitions, exact synonyms, related synonyms, as well as an external reference for each method in form of a PubMed identifier.

Initially, we planned to measure the mean reciprocal rank (MRR) of correctly identified interaction methods (i.e. MI identifiers) for each interaction pair compared to manually curation. Due to the small number of submissions for this task, we finally decided to assess the predictions only considering the precision of extracting associations between documents and the CV of interaction detection methods. The training data for the IMS consisted in a subset of annotations and their corresponding full text articles derived from the IntAct and MINT databases. These articles had been curated manually to extract protein interactions for both the interaction pairs as well as the described interaction detection methods. We recommended for this sub-task not to use articles in the training set describing large scale experiments (i.e. more than 20-30 interactions), because they were excluded also from the test set. Not all the proteins mentioned in a given article are usually studied by all the mentioned protein interaction detection methods.

3.3 Protein Interaction Sentences Sub-task (ISS)

In practice, protein-protein interaction information for a given pair of proteins might be mentioned several times throughout a full text article. To produce a protein interaction summary, for instance, it is useful to select the most relevant sentence expressing interaction information for a given protein pair. Also, for human interpretation, natural language text passages describing a given interaction are useful, especially in case of long full text articles. Therefore, in the interaction sentence sub-task (ISS), we asked participants to provide, for each protein interaction pair, a ranked list of maximum 5 evidence passages describing their interaction. Each passage could contain up to 3 sentences. For the evaluation, pooling methods were used and all the interaction evidence passages (sentences) from all the systems for each document were collected. The predictions were evaluated in terms of percentage of interaction-relevant sentences with respect to the total number of predicted (submitted) sentences and the mean reciprocal rank (MRR) of the ranked list of interaction evidence passages with respect to the manually chosen best interaction sentence.

4 Data collections

For these sub-tasks a larger training collection of full text articles (740) and a smaller collection of test set articles (358) were provided to registered teams. Both collections contained full text articles in different formats, namely as HTML and PDF. Additionally, we also provided these articles as plain text automatically converted from HTML to plain text using html2text and from PDF to plain text using pdftotext. Both collections consisted of subsets of the original training and test set provided by the interaction databases after extensive filtering. For the sub-selection process, the following criteria were taken into account:

- Redundancy: duplicate articles which had been annotated by both databases were removed.
- Journal: only articles from publishers which granted the use of articles for this assay could be included.
- Large scale experiments: articles which mentioned large scale experiments were removed
- Full text: only full articles which are currently available both in HTML and PDF formats were included; in case of articles published before 2000, the full text articles were often only available in PDF.
- Format: In some cases, the articles could not be converted to plain text using the previously mentioned tools and had to be removed.

4.1 Training data collections

In case of the provided training package, in addition to the 740 full text articles in the various previously mentioned formats, the associated annotation files for each article in standard PSI-MI format and as flat annotation files were available to the participants. These annotations contained the normalized interaction pairs, the interaction detection methods, as well as some additional information curated by the interaction databases. Also, a file which comprised the Molecular Interaction (MI) identifiers of concepts which are children or ancestor nodes of the interaction detection method (a total of 155 concepts) formed part of the training package for the IMS.

In case of the ISS, only a limited amount of unique full text interaction evidence passages could be provided for the training collection (63). In compensation, additional resources had been included in the training package:

Anne-Lise Veuthey corpus - a collection of sentences kindly provided by Anne-Lise Veuthey from the Swiss Institute of Bioinformatics (SIB) containing protein interaction related sentences from PubMed abstracts. It has a total of 697 evidence sentences.

Prodisen interaction subset - a collection of 921 sentences related to interactions derived from the Prodisen corpus [Krallinger et al.2006]. Each sentence from a given abstract is manually classified whether it contains interaction descriptions of genes and proteins.

Christine Brun corpus - a set of sentences derived from abstracts related to interactions and their corresponding interaction type (defined as direct or indirect).

GeneRIF interactions - the collection of interaction sentences provided by GeneRIF. There are a total of 51,381 entries in this collection.

Although all of these additional collections are related to interaction sentences, they differ from the passages extracted by the interaction database curators in several points: they are derived from abstracts alone, while the BioCreative interaction evidence passages were extracted from full text articles; they are single sentences while the BioCreative test passages can span several sentences.

4.2 Test data collection

As test set a total of 358 full text articles was provided to the participants. The interaction databases MINT and IntAct had previously curated these articles, but held the derived annotations back until the submission phase of test set predictions was over. These articles were provided in the same formats as the training set and resulted from filtering the initial collection provided by the interaction databases following the sub-selection criteria previously introduced. It was not possible to convert some of the articles to plain text (e.g. PMID 7629138). It was also verified that the overall length and word count of the articles converted to plain text from PDF were consistent with the plain text conversion from the HTML formatted articles.

5 Overview of the used systems

Most of the participating systems did not make use of any additional training data collections for developing their systems, which implies that most of them relied only on the training collections provided by the task organizers. Only few exceptions can be found, for instance in case of team 6, also a proprietary corpus of biomedical papers annotated with proteins and their interactions was used.

In addition to MINT and IntAct also other interaction databases are currently available. The majority of the teams did not exploit annotations derived from other interaction annotation resources.

Some teams had in-house interaction annotation collections, like in case of team 47, which exploited also a collection of their own annotations for the system development.

Most of the participating strategies are characterized by the integration of machine learning techniques to address these sub-tasks, being Support Vector Machines (SVM) the most frequently adapted technique followed by Maximum Entropy (ME) Models.

In order to correctly identify the normalized interactor proteins it is important to associate text mentions to database records (i.e. SwissProt accession numbers). Here the use of protein name tagging and normalization strategies is crucial. The Gene Mention and Gene Normalization tasks of the BioCreative challenge addressed these aspects in the case of PubMed abstracts. For the normalization of the interactor proteins from full text articles, most of the participants used a database look-up and protein name dictionary-based approaches in order to map protein names and symbols contained in the SwissProt database to text mentions. Only few teams made use of more sophisticated protein mention detection methods like LingPipe, Abner, or the Maximum Entropy Markov Model (MEMM) based tagger developed by Curran and Clark.

In full text articles proteins derived from multiple organism sources are often described in the same passage. This is often the case for human proteins and their related mouse homologues. Many protein names contained in biological annotation databases such as SwissProt suffer from inter-species protein name ambiguity, meaning that two proteins from different organism sources share the same name (or symbol). In order to provide correct associations of proteins to SwissProt records, the detection of the corresponding organism source is thus of practical relevance. Surprisingly not all the strategies used for the PPI task applied organism tagging to improve the interactor protein normalization.

Almost all teams integrated currently available NLP components into their systems for these sub-tasks. The most frequently used components were Part of Speech (POS) tagger, stemming and sentence segmentation algorithms as well as tokenization and shallow parsing tools. Some systems also used additional elements, like lemmatization, chunking, and abbreviation extraction (team 6) or predicate analysis (team 49). Among the actual applications used, the following ones could be listed: Brill's POS tagger, MedPost, Stanford parser, Schwartz and Hearst abbreviation extraction tool and MxTerminator for sentence segmentation. Only few teams used external lexical resources such as dictionaries or ontologies. Team 6 exploited for the protein name recognition a proprietary protein listed derived from RefSeq.

A considerable number of strategies were characterized by integrating sentence classifiers to detect interaction-relevant sentences from the full text articles. Another common feature of the participating strategies was the use of regular expressions or pattern matching strategies (for example for the tagging of protein or species names as well as for the interaction detection method identification).

6 Results and evaluation

6.1 IPS results

In practice it is not always possible to normalize mentioned interactor proteins to a single database. This is due to the fact that in general curated databases do not cover all the proteins described in the literature. We thus had to evaluate the interaction pairs submitted by the participants considering also two test set article collections: (a) the set of articles which exclusively mention interaction pairs which can be normalized to SwissProt (referred to as the SwissProt-only article set) and the articles which mention in addition to the SwissProt interaction pairs also interactions where at least one of the interactors could only be normalized to other databases, such as TrEMBL (the whole article set). In general, for the evaluation of text mining tools extracting automatically annotations from the literature, there are two basic evaluation scenarios. This is also true when calculating the performance of interaction pair extraction systems. One evaluation type is based on calculating the global performance of the system compared to the total collection of curated interaction pairs. The other evaluation scenario is based on calculating the average performance for the articles in the test collection. The last approach is actually more useful in practice, as it provides some insight on how stable the method is when applied to a given article.

Team	Run	Precision	Recall	F-score
4	1	0.3893	0.3073	0.2885
6	1	0.2758	0.3011	0.2532
6	2	0.2218	0.2592	0.2066
6	3	0.2392	0.3035	0.2272
11	1	0.0510	0.2753	0.0717
11	2	0.0510	0.2753	0.0717
11	3	0.0517	0.2776	0.0726
14	1	0.1791	0.1421	0.1384
14	2	0.1944	0.1300	0.1414
14	3	0.1162	0.1057	0.0985
17	1	0.0413	0.2543	0.0631
17	2	0.1018	0.2012	0.1182
17	3	0.1633	0.2066	0.1599
19	1	0.0854	0.2115	0.1036
19	2	0.1144	0.2681	0.1361
19	3	0.1595	0.2466	0.1690
28	1	0.1373	0.2905	0.1579
28	2	0.2177	0.2651	0.2039
28	3	0.3096	0.2935	0.2623
30	1	0.0551	0.1888	0.0731
30	2	0.0345	0.2352	0.0528
30	3	0.1574	0.1846	0.1382
36	1	0.0441	0.1121	0.0503
36	2	0.0229	0.0990	0.0305
36	3	0.0548	0.1350	0.0680
40	1	0.0762	0.2489	0.0990
40	2	0.2632	0.2484	0.2171
42	1	0.0160	0.4167	0.0280
42	2	0.2384	0.2218	0.2014
42	3	0.2101	0.2024	0.1827
43	1	0.0395	0.0846	0.0424
43	2	0.0828	0.0680	0.0653
43	3	0.0620	0.0867	0.0592
47	1	0.0830	0.1891	0.0910
47	2	0.0889	0.1909	0.0950
47	3	0.0747	0.1855	0.0844
49	1	0.0109	0.1092	0.0185
49	2	0.0289	0.0557	0.0345
49	3	0.0255	0.0865	0.0357
58	1	0.0003	0.0006	0.0004
58	2	0.0003	0.0006	0.0004
58	3	0.0004	0.0006	0.0005
60	1	0.0323	0.0942	0.0362
60	2	0.0162	0.0558	0.0205
60	3	0.0251	0.0654	0.0299

Table 1: IPS-result all. Average precision, recall and f-score obtained for all the test set articles for each of the evaluated runs of the IPS.

Table 1 contains the results obtained by the IPS participating systems when using the whole test article collection, while table 2 contains the performance when looking only at the set of SwissProt-only articles. All the binary interaction pairs predicted by the participating teams have been compared to the manually-derived interaction pairs extracted from the full text articles.

The corresponding precision, recall and f-score were obtained for each article and then the average was calculated for the whole run. The precision, recall and f-score are defined as follows:

$$Precision = \frac{TP}{TP + FP}; \quad Recall = \frac{TP}{TP + FN}; \quad (1)$$

$$f-score = \frac{2 * Precision * Recall}{Precision + Recall}; \quad (2)$$

where TP: number of True Positive predictions, FP: False Positives, FN: False Negatives, TN: True Negatives, P: total number of Positives and N: total number of Negatives.

Team	Run	Precision	Recall	F-score
4	1	0.3908	0.2970	0.2849
6	1	0.3150	0.3356	0.2871
6	2	0.2519	0.2868	0.2308
6	3	0.2632	0.3394	0.2532
11	1	0.0562	0.2850	0.0770
11	2	0.0562	0.2850	0.0770
11	3	0.0569	0.2879	0.0780
14	1	0.1975	0.1543	0.1510
14	2	0.2113	0.1430	0.1552
14	3	0.1287	0.1157	0.1079
17	1	0.0452	0.2765	0.0684
17	2	0.1138	0.2274	0.1334
17	3	0.1901	0.2396	0.1862
19	1	0.0882	0.2287	0.1092
19	2	0.1200	0.2912	0.1453
19	3	0.1750	0.2748	0.1865
28	1	0.1566	0.3189	0.1784
28	2	0.2434	0.2828	0.2247
28	3	0.3696	0.3268	0.3042
30	1	0.0624	0.2153	0.0824
30	2	0.0367	0.2533	0.0557
30	3	0.1646	0.1964	0.1468
36	1	0.0456	0.1243	0.0560
36	2	0.0202	0.0997	0.0295
36	3	0.0560	0.1362	0.0686
40	1	0.0824	0.2672	0.1083
40	2	0.2751	0.2737	0.2355
42	1	0.0177	0.4368	0.0307
42	2	0.2522	0.2331	0.2112
42	3	0.2278	0.2158	0.1970
43	1	0.0412	0.1032	0.0491
43	2	0.1032	0.0836	0.0803
43	3	0.0734	0.1082	0.0731
47	1	0.0876	0.1964	0.0931
47	2	0.0940	0.1988	0.0978
47	3	0.0791	0.1920	0.0860
49	1	0.0107	0.1085	0.0186
49	2	0.0246	0.0564	0.0319
49	3	0.0234	0.0871	0.0340
58	1	0.0000	0.0000	0.0000
58	2	0.0000	0.0000	0.0000
58	3	0.0000	0.0000	0.0000
60	1	0.0384	0.1113	0.0422
60	2	0.0179	0.0631	0.0213
60	3	0.0281	0.0686	0.0314

Table 2: IPS-result SwissProt only. Average precision, recall and f-score obtained for the SwissProt-only test set articles for each of the evaluated runs of the IPS.

A total of 45 official runs have been received from 16 teams. Most of the teams submitted three runs (the maximum number of allowed runs per team). In general the performance on the set of articles containing only SwissProt protein interaction pairs was higher when compared to the whole test set collection. When considering each of the evaluation scores obtained for the whole test set collection, team 4 obtained the highest average precision (0.39), followed by team 28 (0.31) and team

6 (0.28). In case of recall, team 42 submitted the top performing run (0.42), but with a rather low corresponding precision (0.016). Team 4 obtained a recall of 0.37 and team 6 of 0.30. In case of the average f-scores the single run submitted by team 4 obtained 0.29, followed by run 3 of team 28 (0.26) and run 1 of team 6 (0.25). Refer to figure 1 for the obtained precision-recall plot.

As for the SwissProt-only set of articles, the obtained f-scores were slightly better. Team 28 had the best f-score (0.30), followed by team 6 (0.29) and team 4 (0.28). Table 2 shows the results for each of the teams for this collection of articles. A common characteristic of the top scoring teams was the use of more sophisticated protein mention and normalization strategies, as well as the use of species identification and sentence classification modules when compared to other systems. This implies that one of the main aspects affecting performance of protein interaction extraction systems, in addition to the identification of interaction-relevant sentences is the correct protein mention detection and subsequent normalization.

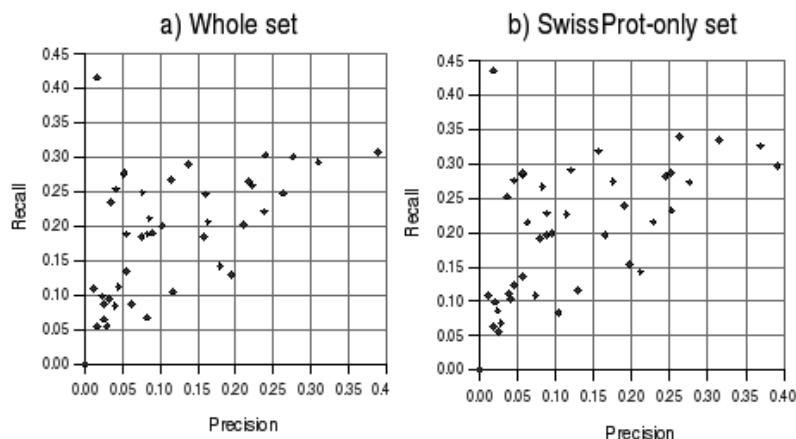


Figure 1: Recall-precision plot of the received runs IPS runs for the whole test set and the SwissProt-only set.

We therefore also evaluated the normalization of the interactor proteins forming each interaction pair. This was done by comparing the list of unique interactor proteins manually curated with the interactors predicted by the participating systems. The average of precision, recall and f-score for articles with at least a single prediction were calculated. Here the highest average precision for the normalization of experimentally characterized interactors was of 0.56 (team 4) in case of the whole collection and of 0.57 (team 28) in case of the SwissProt-only collection. The average recall obtained by each of the evaluated runs was almost always higher than the corresponding precision. In case of the whole test set, team 42 had the highest average recall (0.68) followed by team 30 (0.55) and team 11 (0.54). When looking at the SwissProt-only set, team 42 reached even a recall of 0.69, but with a modest precision (0.08). Finally the highest average f-score for the interactor protein normalization of articles with predictions was of 0.52 (team 28) in case of the SwissProt-only set and of 0.48 (team 4) in case of the whole collection.

6.2 ISS results

The ISS the passages predicted by the participants were pooled, and duplicated passages removed. A unique identifier was assigned to each predicted passage. The same was carried out for the manually extracted best summarizing interaction evidence passages. Then the predicted passages where

compared to the manually extracted ones by sliding the shorter of both over the longer one (previously stripping the HTML tags) and calculating for each position the corresponding string similarity between both. Predicted passages were considered as correct (i.e. mapping to the manually curated ones) in case the string similarity between both was significant. For calculating the string similarity the python difflib library was used. Table three reflects the obtained results. The best result was obtained by team 4, where the fraction of sentences which can be mapped to the manually extracted ones was of 0.19. Unfortunately this team did not follow the submission recommendations for providing the passage rank. This team submitted few passages, but with a high fraction of correct passages. Note that we evaluated here the mapping to the best passages summarizing the interactions, which implies that also alternative sentences could appear in the full text articles describing interactions, but which had not been extracted by the curators.

Team	Run	Total	TP	Unique	TP (Unique)	Perc. Correct	Perc. Correct (unique)	MRR
4	1	372	51	361	51	0.1371	0.1413	-
4	2	372	71	361	70	0.1909	0.1939	-
6	1	2497	147	2072	117	0.0589	0.0565	0.5525
11	1	18385	360	5156	131	0.0196	0.0254	0.6594
11	2	18371	376	5270	145	0.0205	0.0275	0.6253
11	3	18371	387	5252	156	0.0211	0.0297	0.6416
14	1	634	13	579	12	0.0205	0.0207	0.8718
14	2	458	10	422	10	0.0218	0.0237	0.8167
14	3	560	13	514	11	0.0232	0.0214	0.8718
27	1	1420	37	1386	36	0.0261	0.0260	0.4653
28	1	3028	150	3001	148	0.0495	0.0493	0.3740
28	2	2249	127	2231	126	0.0565	0.0565	0.3696
28	3	5448	352	3210	191	0.0646	0.0595	0.3392
36	1	4515	232	3407	169	0.0514	0.0496	0.5731
36	2	11827	571	7526	343	0.0483	0.0456	0.5813
36	3	4083	247	3018	161	0.0605	0.0533	0.5476
43	1	3691	111	3117	97	0.0301	0.0311	0.4083
43	2	1507	69	1383	63	0.0458	0.0456	0.3524
43	3	3674	148	3257	131	0.0403	0.0402	0.3449
47	1	7934	278	4975	159	0.0350	0.0320	0.5232
47	2	7633	274	4835	156	0.0359	0.0323	0.5205
47	3	8355	290	5172	163	0.0347	0.0315	0.5329
49	1	21431	590	10422	285	0.0275	0.0273	0.3785
60	1	2243	104	2019	91	0.0464	0.0451	0.3460
60	2	4714	157	3932	130	0.0333	0.0331	0.3959
60	3	7780	229	6293	192	0.0294	0.0305	0.3998

Table 3: ISS-result. This table reflects the baseline evaluation of the submissions received for ISS. Here the submitted passages were compared to the previously manually selected passages reflecting the best interaction evidence. TOTAL: total number of evaluated passages (note that submissions of articles for which the curators could not find a suitable evidence passage were excluded from evaluation). TP: number of correct passages (i.e. mapping the manually annotated ones). Unique: number of unique passages per run (after removing duplicate passages). TP(Unique): number of correct passages (i.e. mapping the manually annotated ones) in the collection of unique passages. Perc. Correct: fraction of predicted passages corresponding to the 'best' previously extracted passages. Perc. Correct unique: fraction of unique predicted passages corresponding to the 'best' previously extracted passages, MRR: mean reciprocal rank of the correct passages. Note that in case of team 4, the MRR should not be taken into account, as all the submitted passages here were labeled with rank 1 by this team.

The main differences between the predicted passages compared to the previously manually extracted ones was that the latter often also mention the experimental interaction method or a reference to figures where the experimental outcome is shown, while the former do not. This is partially due to the limited amount of full text training sentences and the fact that abstract-derived interaction sentences provided in the additional material collection often lack the experimental characterization mentions. For a sample sentence extracted by the curators which reflect this aspect see:

HAX-1 co-immunoprecipitates with BSEP, MDR1, and MDR2 from transfected cells and hepatocytes.

Here co-immunoprecipitates implies that an co-immunoprecipitation experiment was done which

confirmed the interaction between HAX-1 and BSEP, MDR1, and MDR2

The ISS top performing team actually took into account most of the potentially relevant aspects: the location of the sentence in the document, the relation with figures and tables, whether interaction-indicating keywords were present, the mention of experimental methods as well as summary-indicating cue words.

6.3 IMS results

In case of the IMS only two teams submitted predictions, one of them also provided three additional runs to be evaluated out of the official contest. A total of 874 associations of articles and molecular interaction detection method concept were provided in the test set. Each submitted prediction was evaluated based on three different approaches, using the gold standard of manually extracted annotations as evaluation basis. All PubMed articles were annotated using Molecular Interaction (MI) ontology controlled vocabulary terms (characterized by their Interaction Method identifiers). They are child terms of the "Interaction Detection Method" (ID=MI:0001) subtree (PSI-MI Ontology file: "psi-mi25.obo", dated "12:05:2006 08:47"). The evaluation approaches were basically:

- Exact Matching: if the predicted interaction method ID is an exact match with respect to the gold-standard annotated MI IDs for each PubMed ID.
- Parent Matching: if the predicted interaction method ID is either an exact match of an annotated MI ID for that PubMed ID or if it is a parent ID of one of the annotated MI IDs. A parent was defined as upper node (a parent, more general concept) in the IM-subtree of the MI Ontology and directly related to it. This is done by setting the IM-subtree of the PSI-MI25 Ontology up as a DAG (directed acyclic graph) and testing if there is a path from the predicted ID to the annotated ID.

Team	Run	Precision	Recall	F-Score
14	1	0.3628	0.2172	0.2513
14	2	0.3186	0.1980	0.2249
14	3	0.3348	0.1938	0.2265
40	1	0.6679	0.3383	0.4207
40	2	0.4028	0.5548	0.4363
40	3	0.5068	0.5222	0.4836

Table 4: IMS-result: Exact Matching. The results correspond to the averages calculated after scoring each article in terms of precision, recall and f-score for the identification of exact matching of article to normalized Molecular Interaction (MI) identifiers.

Team	Run	Precision	Recall	F-Score
14	1	0.4986	0.3078	0.3495
14	2	0.4471	0.2847	0.3170
14	3	0.4881	0.2953	0.3375
40	1	0.6794	0.3472	0.4302
40	2	0.5899	0.8548	0.6519
40	3	0.6541	0.7093	0.6375

Table 5: IMS-result Parent Matching. Same as table 3 but using the parent matching evaluation

Tables 4 and 5 show the results obtained by teams 14 and 40. The difference between the exact matching and the parent matching results was not as big as expected. This could be partly explained by the fact that the more specific concepts in the MI ontology (child nodes) actually are closely related to the method names as they are used in the literature. The first run of team 40 obtained the best precision (0.67), while the best f-score corresponded to run 3 of the same team (0.45). The underlying

approach is characterized by the use of pattern matching, the automatic generation of well-known variants of the method names provided in the MI ontology and the generation of handcrafted patterns for some of the methods.

7 Discussion and Conclusions

The Protein-Protein Interaction task of the second BioCreative challenge tried to cover all the main aspects relevant for automatically extracting biological annotations from the scientific literature, namely for normalized and experimentally verified protein interactions. It also reflects the importance of collaborative efforts between domain experts, which manually curate biological relevant information from the literature, and the text mining community.

The average performances, on full text articles, of participating systems, as well as the limitations when using text mining techniques to recover such interactions have been explored. Although the initial results are promising, they also indicate that certain components still need further improvements and have currently not been sufficiently taken into account. Several obstacles were encountered by the participating systems that increased the difficulty in detecting normalized interaction pairs from full text articles. To name a few that had an influence on the obtained results refer to the list below:

- Errors resulting from conversion of PDF or HTML formatted documents to plain text, such as page break errors, wrong special character handling and word joining.
- Sentence boundary detection errors and difficulties in processing tables and figure legends.
- Multiple organism mentions and the resulting inter-species ambiguity for protein normalization.
- Incompleteness of currently available protein normalization resources. Existing annotation databases such as SwissProt do not contain all the symbols or names for described proteins in the literature.
- Difficulty in association extraction and coordination handling of multiple interaction pairs from a single sentence.
- Interaction evidence phrases in legends or titles which often do not correspond to grammatically correct sentences.
- Heavy use of domain specific terminology, for instance in case of experimental descriptions.
- Dispersed interaction evidences contained in not consecutive sentences.
- The use of domain expert inference and bioinformatics tools to perform protein normalization in order to normalize some of the interactors.
- Errors in shallow parsing and POS-tagging tools trained on general English text collections when applied to biomedical texts.

It is well known that large training and test collections of full text articles with in depth annotations of biological relevant information useful also for developing text mining systems will improve the performance of the current technologies. The data collections derived from this BioCreative PPI task can be seen as one more contribution in this direction and will be released after the evaluation workshop as a useful resource for evaluating protein interaction extractions compared to manual curation. This is also true for protein mention and normalization components, where the BioCreAtIvE challenge has already provided useful resources for abstract processing [Blaschke et al.2005, Yeh et al.2005, Hirschman et al.2005]

As a general observation on the outcome of the strategies used, it can be stated that the most sophisticated and complete systems did outperform significantly more basic strategies, which often only adapted for this task existing supervised learning modules. In case of the top performing teams such as team 4, 6 and 28 both general language as well as domain specific resources were exploited. It is therefore clear that using sophisticated gene mention and normalization detection strategies generally improved the results of participating teams and constitute one of the most important components for interaction extraction systems. Also efficiently handling linguistic coordination is crucial when extracting associations such as protein-protein interactions. The use of supervised-learning-based sentence classifier and the detection of interaction method names also seemed to play a role in the performance of interaction detection strategies.

When comparing the performance of the interactor protein normalization and the interaction pair extraction, it seems that the extraction of the interaction pairs is slightly better than would be expected, even if details are still unclear it might indicate gain from global information contained in the articles.

One aspect which had not been addressed in the current BioCreative edition is how the resulting systems would perform when doing interactive evaluation as part of curation-assistance tools. To close the gap between text mining systems and the actual end users such interactive assessments would be especially useful. Here aspects such as interaction ranking, and time spent per curation when using the text mining systems compared to baseline PubMed search-based approaches, could provide additional insights on the importance of literature mining applied to the biomedical domain.

8 Acknowledgements

The BioCreative PPI task would not have been possible without the collaborations of the MINT and IntAct databases, which provided the full text article curations, to Lynette Hirschman and Carlos Rodriguez for useful feedback and comments and to the participants for their great effort in developing protein-interaction extraction systems and taking part in this challenge. Many thanks also to the publishers for granting the use of the full text versions of the used articles for the BioCreative challenge, especially to Nature Publishing Group and Elsevier and their consequent support to the text mining community.

References

- [Blaschke and Valencia2002] C. Blaschke and A. Valencia. 2002. The frame-based module of the Suiseki information extraction system. *IEEE Intelligent Systems.*, 17:14–20.
- [Blaschke et al.2005] C. Blaschke, E. Andres Leon, M. Krallinger, and A. Valencia. 2005. Evaluation of BioCreative assessment of task 2. *BMC Bioinformatics.*, 6:S16.
- [Daraselia et al.2004] N. Daraselia, A. Yuryev, S. Egorov, S. Novichkova, A. Nikitin, and I. Mazo. 2004. Extracting human protein interactions from MEDLINE using a full-sentence parser. *Bioinformatics.*, 20:604–611.
- [Hermjakob et al.2004] H. Hermjakob, L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman, and R. Apweiler. 2004. IntAct: an open source molecular interaction database. *Nucleic Acids Res.*, 32:D452–D455.
- [Hirschman et al.2005] L. Hirschman, M. Colosimo, A. Morgan, and A. Yeh. 2005. Overview of BioCreAtIVe task 1B: normalized gene lists. *BMC Bioinformatics.*, 6:S11.

- [Krallinger et al.2006] M. Krallinger, R. Malik, and A. Valencia. 2006. Text Mining and Protein Annotations: the Construction and Use of Protein Description Sentences . *Genome Inform Ser Workshop Genome Inform*, 17:121–130.
- [Ono et al.1999] T. Ono, A. Tanigami, H. Hishigaki, and T. Takagi. 1999. Automatic extraction of information on protein-protein interaction from scientific literature. *Proc GIW 99*.
- [Orchard et al.2005] S. Orchard, L. Montecchi-Palazzi, H. Hermjakob, and R. Apweiler. 2005. The use of common ontologies and controlled vocabularies to enable data exchange and deposition for complex proteomic experiments. *Pac Symp Biocomput.*, pages 186–196.
- [Uetz et al.2000] P. Uetz, L. Giot, G. Cagney, T.A. Mansfield, R.S. Judson, J.R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J.M. Rothberg. 2000. A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, 403:623–627.
- [Valencia and Pazos2002] A. Valencia and F. Pazos. 2002. Prediction of protein-protein interactions from evolutionary information. *Methods Biochem Anal.*, 44:411–426.
- [Yeh et al.2005] A. Yeh, A. Morgan, M. Colosimo, and L. Hirschman. 2005. BioCreAtIvE Task 1A: gene mention finding evaluation. *BMC Bioinformatics.*, 6:S2.
- [Zanzoni et al.2002] L. Zanzoni, A. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni. 2002. MINT: a Molecular INTeraction database. *FEBS Lett*, 513:135–140.



Annotating molecular interactions in the MINT database

Andrew Chatr-aryamontri¹
aryamontri@hotmail.com

Arnaud Ceol¹
Arnaud.ceol@uniroma2.it

Luana Licata¹
Luana.Licata@gmail.com

Gianni Cesareni¹
cesareni@uniroma2.it

¹ Department of Biology, University of Rome, Tor Vergata, Via della Ricerca Scientifica, 00133 Rome Italy

Abstract

The Molecular INTeraction Database (MINT, [6]) is a relational database storing protein-protein interactions. We give here a highlight on aspects of the curation procedure that are also relevant for the evaluation of the Biocreative competition results.

Keywords: MINT, database, protein interaction

Introduction

The post-genomic era has seen the explosion of biomedical databases available on the internet through user friendly web applications. This proliferation is motivated by the need for easy retrieval of information which is otherwise dispersed in a text format throughout the literature and for effective management of the huge datasets generated by high-throughput technologies.

The Molecular INTeraction Database (MINT, [6]) was conceived with the aim of storing experimentally verified protein-protein interactions published in peer-reviewed journals.

Over the past years, in order to exhaustively and accurately represent all aspects of molecular interactions, MINT has undergone continuous upgrades of both the database structure and the curation procedure [1]. This included the adoption of the IntAct relational model [3], an open source project specifically developed for the storage and analysis of molecular interactions, and of the HUPO PSI molecular interaction format level 2.5. We will briefly describe here the curation rules and procedures that are relevant for the Biocreative competition.

Curation: standards and rules

Syntax and semantics for data representation in MINT are provided by the Proteomics Standards Initiative-Molecular Interaction (PSI-MI 2.5) standards as established by the PSI-MI workgroup, of which MINT is an active member [2]. This workgroup develops and maintains a common data model for the representation and exchange of interaction data. The schema and the controlled vocabularies (CVs), which allow representation of binary and n-nary interactions, are continually updated to permit increasingly accurate descriptions of molecular interactions.

Interaction records in MINT represent either physical interactions or co-localizations (fig.1) in accordance with the PSI-MI standards, where “physical interactions” are defined as “interactions among molecules that can be direct or indirect”. Since genetic interactions describe functional relationship among genes they are considered distinct from physical interactions between proteins and are not currently curated by MINT.

Each entry describes an interaction, its participants and the experimental procedure used to discover or to prove the interaction.

Since MINT stores experimentally verified interactions, and not all experimental methods are equally reliable, strong emphasis is put on thorough description of the experimental evidence. This allows users to filter the data and apply their own confidence values. Submission pages prompt the curator for the required information, allowing a rich annotation of the features of the experiment and the participants. The minimal information required for entering an interaction in MINT are the PubMed identifier (pmid) of the publication demonstrating the interaction and the Uniprot accession numbers of the interacting proteins. For each participant it is possible to annotate a number of features: experimental role, biological role, expression level, sample process, tags and identification method, as per the PSI-MI 2.5 schema and CVs. Furthermore, whenever the information is available, it is possible to describe the protein region involved in the interaction (as a binding site range) and to cross-reference this binding site to InterPro. Other participant features that can be annotated include mutations and modifications shown to affect the interaction strength, and whether the impact of the modification was found to be positive or negative (fig.2).

In the experiment description form, the curator reports the experimental method that was used to detect the interaction (such as yeast 2-hybrid, pull down, or co-immunoprecipitation), the interaction type (physical interaction, co-localization) and where the interaction was observed (organism or *in vitro*). Whenever provided in the source publication, this description also includes the kinetic constants of the interaction and any author-supplied confidence value (fig.3).

In order to ensure the fidelity of the curation process MINT uses two different quality control systems. A first control is an automated one, based on curation rules, which ensures that mandatory fields are filled and that annotated ranges or residues are consistent with the protein length reported by Uniprot. In addition, every new entry undergoes a validation step performed by a second curator before it is released to the public database. Each entry provides the number of the figure or the table reporting the interaction. This allows the second curator to check quickly if the reported information is consistent with the described experiment, paying particular attention to the correctness of the uniprot identifiers.

Curation: projects

The MINT curation team is composed of PhD level curators: two full-time and one part-time curators. Each curator undergoes training in the database standards that allows them to fully curate, in an accurate and consistent fashion, the literature describing interactions derived from low-throughput experiments.

MINT regularly curates new issues of FEBS Letters (since January 2005), EMBO Journal and EMBO Reports (both since January 2006). This choice was made in agreement with the other members of the International Molecular-Interaction Exchange consortium (IMEx; [7]), currently including DIP [5] and IntAct [4]. The IMEx agreement aims to avoid work overlaps, to share the curation workload and to exchange curated molecular interaction (MI) data. All IMEx members share common curation rules as described in the reference manual available at the IMEx web site [8]. In addition to the IMEx-specified curation commitments, MINT focuses on curating papers describing interactions mediated by protein domains and viral proteins.

Curation: hurdles

The first major hurdle to flawless curation remains the identification of suitable source publications. In order to assess a paper for the presence of “curatable” interaction data, curators quickly read the title and the abstract. Although this approach is adequate for the vast majority of papers, it is still possible that papers with interaction data are missed this way. The “thorough” curator also inspects all the figures and the Materials and Methods section.

Once a paper is identified for curation the most critical point resides in the identification of the interacting molecules. Up to 70% of overall curation time can be spent on mapping molecule identifiers unambiguously to well-characterized database entries. Often the author describes the protein only as “mammalian”, making it impossible to unambiguously identify which mammalian genome the protein is encoded from. In

some cases the authors refer to the name of a protein complex without specifying which subunit has been used in the experiment. For instance it is not sufficiently accurate to write that the 14-3-3 protein interacts with protein B since seven 14-3-3 isoforms are encoded in a mammalian genome. On average the curation of a manuscript describing interaction data takes up to two-three hours work of an expert curator, thus setting the curation rate at about 3-4 papers a day.

MINT contributions to Biocreative

MINT provided two different datasets to the Biocreative competition, from which a test set and part of the training set were compiled. The training set was mainly composed of papers already curated and publicly released (table I). The MINT/BioCreative test set was made of papers extracted from volumes of FEBS letters, EMBO Journal and EMBO Reports published between January 2006 and July 2006. The curated articles belong to the positive test set while the ones that the curator assessed as not relevant form the negative test set. The public release of the MINT entries derived from the curation of the above-mentioned issues was therefore delayed till the end of the competition. As an additional task specifically for the Biocreative test set, the curators were asked to identify and report the best sentence describing the interaction, This was extracted from either the body text or the figure legends, t

Potential pitfalls of protein interactions prediction by text mining

Here we describe a list of potential problems in the curation process that might affect Biocreative predictions.

Although MINT curators are expert and thorough biologists, they are humans and occasionally they make mistakes. Thus, it can happen that some of the entries contain errors.

False negatives

It is not always possible or easy to identify a single sentence that clearly describes an interaction reported in a paper. In many cases the evidence that a paper is “curatable” is dispersed throughout multiple sentences in the full text article (eg: “the two proteins co-purify together”). Nevertheless, curators can clearly identify and extract an interaction from a figure or a table, even if there is no sentence explicitly reporting that interaction in the text. For instance, positive controls are not usually cited in the text and interactions from high-throughput experiments are reported in tables.

False positives

For text miners the presence of the word “interaction” in the text directly points to an interaction. Unfortunately the “interaction” can refer to experiments describing genetic interactions which are not curated by MINT, to drug-drug interactions, or to other data irrelevant to MINT. In other cases there is no experimental evidence supporting the interaction

Interactions mediated by complexes

Interactions between protein complexes (eg: Pol II) and proteins are not considered by MINT curators. In these cases, the interactions detected by the text-mining tool will not find any match in MINT records.

References

- [1] Chatr-aryamontri, A., Ceol, A., Palazzi, L.M., Nardelli, G., Schneider, M.V., Castagnoli, L., Cesareni, G., MINT: the Molecular INTeraction database. *Nucleic Acids Res.* 35(Database issue):D572-4, 2007.
- [2] Hermjakob, H., Montecchi-Palazzi, L., Bader, G., Wojcik, J., Salwinski, L., Ceol, A., Moore, S., Orchard, S., Sarkans, U., von Mering, C., Roechert, B., Poux, S., Jung, E., Mersch, H., Kersey, P., Lappe, M., Li,

- Y., Zeng, R., Rana, D., Nikolski, M., Husi, H., Brun, C., Shanker, K., Grant, S.G., Sander, C., Bork, P., Zhu, W., Pandey, A., Brazma, A., Jacq, B., Vidal, M., Sherman, D., Legrain, P., Cesareni, G., Xenarios, I., Eisenberg, D., Steipe, B., Hogue, C., Apweiler, R., The HUPO PSI's molecular interaction format--a community standard for the representation of protein interaction data. *Nat. Biotechnology* 22(2):177-83, 2004.
- [3] Hermjakob, H., Montecchi-Palazzi, L., Lewington, C., Mudali, S., Kerrien, S., Orchard, S., Vingron, M., Roechert, B., Roepstorff, P., Valencia, A., Margalit, H., Armstrong, J., Bairoch, A., Cesareni, G., Sherman, D., Apweiler, R., IntAct: an open source molecular interaction database. *Nucleic Acids Res.* 32(Database issue):D452-5, 2004.
- [4] Kerrien, S., Alam-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., Dimmer, E., Feuermann, M., Friedrichsen, A., Huntley, R., Kohler, C., Khadake, J., Leroy, C., Liban, A., Liefink, C., Montecchi-Palazzi, L., Orchard, S., Risso, J., Robbe, K., Roechert, B., Thorneycroft, D., Zhang, Y., Apweiler, R., Hermjakob, H., IntAct--open source resource for molecular interaction data. *Nucleic Acids Res.* 35(Database issue):D561-5, 2007.
- [5] Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U., Eisenberg, D., The Database of Interacting Proteins: 2004 update. *Nucleic Acids Res.* 32(Database issue):D449-51, 2004.
- [6] <http://mint.bio.uniroma2.it/mint/>
- [7] <http://imex.sourceforge.net/>
- [8] <http://imex.sourceforge.net/doc/imex-curationManual.doc>

FIGURES

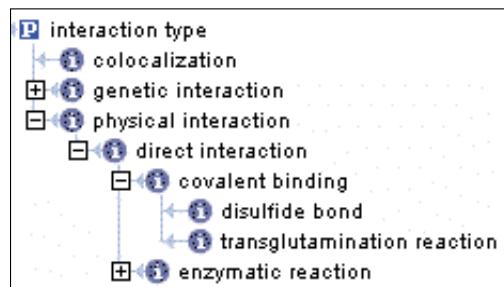


Fig1: Interaction type

back page: 1

CREB-binding protein	cbp (Q92793) 2442 AA	sample process	purified
Homo sapiens (9606)	CREBBP, CBP,	expression level	
Author assigned name:	CBP	tag	gst tagged
biological role	unspecified	n-terminal	bait
interactor type		identification	nucleotide sequence identification
Binding site: experimentally detected:		inferred:	
Range 1069-1892	sequence:	Range	sequence:
identification method:	deletion analysis	identification method:	
<input type="checkbox"/> IPR000433/Znf_ZZ	<input type="checkbox"/> IPR000197/Znf_TAZ	<input type="checkbox"/> IPR009255/Trans_coact	<input type="checkbox"/> IPR003101/kOX
<input type="checkbox"/> IPR010303/DUF902_CREBbp	<input type="checkbox"/> IPR001487/Bromodomain	<input type="checkbox"/> IPR000433/Znf_ZZ	<input type="checkbox"/> IPR000197/Znf_TAZ
other term:		other term:	
<input checked="" type="radio"/> complete <input type="radio"/> add modification <input type="radio"/> add mutation <input type="button" value="continue"/>			

Fig2: interactor submission form

back page: 1

Experiment:	11438528		
pmid:	11438528	detection method:	gst pull down
label:	cho-2001	methods (other pmids):	<input type="text"/> mandatory field
flags (FEBS, EMBO, SH3...):		URL:	
Biosource			
Organism (taxid):	-1	tissue:	
Interaction			
interaction type:	physical interaction	confidence measure:	
negative interaction:	false	confidence value:	
kinetics:	kd	mM	figures: F1B mandatory field
Other annotations (confidences...)			
Additional comments (new lines will be automatically replaced by)			
Curators comments those comments are only seen by curators and are hidden in search pages (new lines will be automatically replaced by)			
<input checked="" type="radio"/> complete <input type="radio"/> add PTM <input type="radio"/> add annotation <input type="button" value="continue"/>			

Fig3: experiment submission form



IntAct - Serving the text-mining community with high quality molecular interaction data

Jyoti Khadake*; Bruno Aranda; Cathy Derow; Rachael Huntley; Samuel Kerrien; Catherine Leroy; Sandra Orchard; Rolf Apweiler; Henning Hermjakob

EMBL – European Bioinformatics Institute, Wellcome Trust Genome Campus, Cambridge, CB10 1SD, UK

Email Addresses:

Jyoti Khadake jyoti@ebi.ac.uk
Bruno Aranda baranda@ebi.ac.uk
Cathy Derow cderow@ebi.ac.uk
Rachael Huntley huntley@ebi.ac.uk
Samuel Kerrien skerrien@ebi.ac.uk
Catherine Leroy cleroy@ebi.ac.uk
Sandra Orchard orchard@ebi.ac.uk
Rolf Apweiler apweiler@ebi.ac.uk
Henning Hermjakob hhe@ebi.ac.uk

* To whom correspondence should be addressed.

Abstract:

Background

IntAct provides an open source database and toolkit for the storage, presentation and analysis of molecular interactions. High quality manual annotation of the literature is a time consuming process and coverage of the available interaction data is far from complete. The use of text-mining procedures to highlight appropriate publications and make an initial extraction of interaction data could help to improve both the efficiency of the curation process and the reporting of the data available in the literature. The 2006 BioCreative competition was aimed at evaluating the success of such procedures in comparison to manual annotation.

Results

To aid the BioCreative protein-protein interaction task, IntAct [1] together with the MINT [2] database, provided both the training and the test datasets. During the manual curation process, the major cause of data loss in mining the articles for information was ambiguity in the mapping of the gene names to the stable UniProtKB database identifiers. It was also observed that most of the information about interactions was contained within the full text of the publication; hence, text-mining of protein-protein interaction data will require the analysis of the full text of the articles and cannot be restricted to the abstract.

Conclusion

The development of text-mining tools to extract protein-protein interaction information may increase the literature coverage achieved by manual curation. To support the text-mining community, IntAct provides the sentences from the articles describing the interactions. These will supply data-miners with a high quality dataset for algorithm development. The dictionary of terms created by the competitors could help enrich the controlled vocabulary synonym list.

Background

An important step in functional systems biology is the understanding of the relationships between biomolecules. Interactions between proteins are crucial to biological pathways. The knowledge of the processes in which the proteins are involved is essential for a fundamental understanding of the cellular machinery. The IntAct database (<http://www.ebi.ac.uk/intact>) [1] is a repository for manually curated molecular interaction data, predominantly related to protein-protein interactions. IntAct aims to capture a full representation of the interaction data available in the literature but this is a time-consuming process and made more difficult by a number of factors. Firstly, the rate at which data is being produced is increasing steadily. This is due to an increased use of high throughput techniques for the detection of protein interactions. Secondly, many authors continue to use ambiguous gene or protein

names in publications or fail to identify the organism from which the gene(s) or protein(s) originate. The failure to provide this information results in a high percentage of the workload of an annotator is the gathering of this information from References, Supplemental Materials, websites and through communications with authors. This has been recently addressed in the MIMIx recommendations to be published in Nature Biotechnology [3]. Once identified, these proteins then have to be correctly mapped to a high quality protein sequence database such as UniProtKB [4], which provides a common platform allowing the management of data redundancy and updates. These factors combine to slow down the manual curation process and prevent databases from achieving their required goal of complete literature coverage.

The literature describes a range of experimental techniques, which can be used to detect the different types of interactions. One of the most important advances in interaction data annotation, querying and exchange is the development by the Molecular Interactions (PSI-MI) work group of the Human Proteome Organisation Proteomics Standards Initiative (HUPO-PSI) [5] of a standardized, hierarchical, ontology of terms used for describing accurately interaction data, the PSI-MI controlled vocabulary (CV) [6]. This may be viewed in the Ontology Lookup Service (<http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=MI>) [7]. IntAct exhaustively uses the PSI-MI CV terms so as to describe interaction data consistently. Advances in the techniques for determining and describing interactions are reflected in the continual evolution of the CV. A snapshot of the hierarchical PSI-MI CV can be seen in Figure 1. A part of the curation process involves determining the exact term describing the methodology used by authors to detect the interactions. The data itself is stored and disseminated using the *de facto* standard: PSI-MI XML2.5 [8].

Due to the accumulation of interaction data in biomedical literature, the need for text-mining tools to facilitate the extraction of such information is urgent. The development of effective text-mining tools could aid the mapping of protein interactors to their UniProtKB identifiers as well as selecting the text, which describes the interaction, and matching these to the PSI-MI CV. This could complement manual curation by speeding up the information extraction process, thus increasing literature coverage. The BioCreative protein-protein interaction task addresses precisely these issues. In order to make text-mining tools useable in real world scenarios, for instance to assist database curators, comparisons and evaluations of different approaches to text-mining are necessary.

To assist with the BioCreative protein-protein interaction task, the IntAct database has contributed both a training set for development of algorithms and a test set for evaluation of the text-mining tools. The IntAct database contribution to the test set was an initial collection of protein-protein interactions extracted from 154 full-text articles. These were then evaluated for suitability and most were used to generate the final test set provided to the participating teams by the organizers. The data from these articles were made publicly available after the completion of BioCreative sub-tasks. Here we give a perspective on the curation process, explain how we chose the papers, extracted the information, manually annotated IntAct entries and the checking process followed to ensure data consistency and quality. We have also described specific annotations on entries introduced to aid the text-mining community and discussed some of the problems encountered during the BioCreative curation effort.

Results and Discussion

IntAct database contribution to the BioCreative training set:

Protein-protein interaction information extracted from articles during the years 2005 and early 2006 formed the contribution of the IntAct database to the training dataset. There was no pre-selection of particular journals within this set. The data was made available in the PSI-MI XML2.5.

IntAct database contribution to the BioCreative test set:

The perusal of the abstracts, and in a few cases a rapid survey of the full-text article, from 6 Journal of Biological Chemistry (JBC) issues resulted in a total of 131 candidate articles. A detailed full-text assessment indicated that 17 of these 131 articles could not be curated into IntAct for the following reasons:

1. The gene names or identifiers described in the article could not be mapped to UniProtKB entries. This was due to an ambiguous description of the gene name, species, subtype of the protein in question or more rarely the absence of a UniProtKB entry for the molecule involved in the interaction. This was the major cause of loss of data resulting in 12 of the 17 articles not being entered.

2. The articles referred to modeling studies, mutation analysis or siRNA studies. In IntAct only an 'interaction detection method' which is a child term of the root terms 'biophysical', 'protein complementation assay', 'biochemical' and 'imaging techniques' in the PSI-MI CV is added to the database. This resulted in a loss of 3 articles.

3. The articles reported genetic or predicted interactions. In IntAct only an 'interaction type' which is defined as a child term of 'physical interaction' or 'colocalization' in the PSI-MI CV is curated. This resulted in a further loss of 2 articles.

None of the problems listed above could be identified by reading the abstract in isolation. It is thus important to note that a full-text analysis is necessary to extract all the interaction data present in literature. A further 40 papers were also curated from other issues of JBC and the journals belonging to the Nature Group of Publishers to complete the set.

Contribution to the text-mining community:

Information, which is present in the article and relevant to an interaction but cannot be fully described by the PSI-MI CV, is added to the IntAct records using additional annotations on the entries. An annotation topic 'dataset' with the description 'BioCreative - Critical Assessment of Information Extraction systems in Biology' was introduced to identify the entries that contributed to the BioCreative test set. 154 articles involving 484 experiments were tagged with this annotation.

In order to aid the text-mining community in identifying the protein interaction sentences from the curated article, an annotation topic 'source-text' was introduced in IntAct. Overall, 815 'source-text' annotations were added to 951 interactions in the BioCreative test set prepared by the IntAct database (see Table 1). The normalized protein interaction sentences generated from the BioCreative initiative will be made available by the organizers.

Since the BioCreative initiative, IntAct database has continued to extract the protein interaction sentences. 3259 'source-text' annotations were added to 3267 interactions as of 31st March 2007. We store about one 'source-text' annotation per interaction. The PubMed ID, IntAct interaction accession number and the 'interaction sentence' are available for download from IntAct via the FTP site: <ftp://ftp.ebi.ac.uk/pub/databases/intact/current/various/data-mining>

Text-mining and the development of the PSI-MI controlled vocabulary:

PSI-MI controlled vocabularies provide a consistent set of terms used to annotate the interaction data. The vocabulary is continually updated to assimilate the newer and more sophisticated techniques. Synonyms for each term are stored within the CV to assist the user in finding the appropriate expression. The dictionary of synonyms developed by the text-mining community both during the competition and in the future, could be incorporated and greatly enhance the information stored on these records.

Manual curation is laborious; the process of curating a paper will on average take up to a day of a trained curator's time, much of which is consumed in adding significant value to the interactions. Initial identification of the interactors and interaction technique is then followed by an in depth analysis of the interactors and the interactions. The PSI-MI CV is used extensively to define the interactors and interactions. InterPro signatures [9] and GO terms [10] are also used to further enrich the information provided to users. The additional steps ensure full and accurate data representation. Other data extracted from literature during the manual curation process is described in Additional Materials.

Conclusions

IntAct provides high quality and well-documented interaction data from the literature using a controlled vocabulary, which reduces the ambiguity surrounding the naming of the techniques. This is achieved through careful manual curation by highly qualified curators. However, as both the volume of literature and the number of proteins requiring characterization increases, the manual processing capability can become overloaded. Semi-automated assistance would greatly expedite the curation process. Text-mining in the biomedical domain is receiving increasing attention. To aid and encourage the development of such tools, the IntAct team at the European Bioinformatics Institute agreed to take part in the BioCreative protein-protein interaction challenge. IntAct contributed to the training set, which can be used to develop the text-mining process and the test set which can be used for the evaluation of the competitors' results.

The interactions themselves are not described in sufficient detail within an article abstract alone, as was demonstrated by the publications that could not be curated from the selected abstracts. This highlights

the importance of the full-text text-mining process. This is necessary for both the identification of interactors as well as description of the interaction.

Manual literature mining can extract more detailed interaction data than is possible by text-mining, and more accurately define the interactors and the interactions. A critical step in literature mining is mapping biological entities to entries in public domain databases such as UniProtKB for proteins. This may require the mapping of highly ambiguous and multiple gene/protein names. Automated mapping of the proteins to UniProtKB entries and the extraction of ‘interaction detection method’ from the articles would improve the literature coverage and efficiency of the manual curation process.

As a commitment to the text-mining community, IntAct continues to provide the sentences used for identifying the interactions under the annotation topic ‘source-text’. A continued interaction between the two communities is necessary to develop an effective text-mining solution to the problems of automated interaction data extraction from published articles.

Materials and methods

The IntAct database contribution to the BioCreative protein-protein interaction task was divided according to the various subtasks of the competition. Curation of entries from PubMed articles was carried out by IntAct to assist with the BioCreative task [11]. The data was curated *as per* the Annotation Manual available at <http://www.ebi.ac.uk/~intact/site/doc/IntActAnnotationRules.pdf>. Additional information pertaining to the interaction that cannot be described using the PSI-MI CV terms is stored in IntAct as annotations on the entry. A publication may report one or more experimental methods, each of which may have one or more interactions.

Determination of the training set:

IntAct database contribution to the training set consisted of data from the articles curated during the years 2005 and early 2006.

Protein interaction subtask IAS - Choosing the articles for the test set:

An important initial exercise was to select the articles to be curated. This is essential, since not all published articles describe protein-protein interactions. The BioCreative competition committee provided a list of journals available for the curation task. Initial articles were initially chosen from JBC issues released on 6th, 13th, 20th and 27th of January 2006 and 3rd and 10th of February 2006 by perusing the article abstracts manually and in some cases a rapid reading of the full-text paper for interaction information. Forty articles were also curated from other JBC issues or the journals belonging to the Nature Group of Publishers. The information available within the full-text and ‘Supplementary Material’ of appropriate articles was manually curated into the IntAct database. The rest of the articles from these 6 issues of JBC were classified as not relevant for this task and served as a negative control.

Protein interaction pairs subtask IPS - Mapping of the interactors to the UniProtKB proteins:

The full text of the article often contained sufficient details to allow the identification of the UniProtKB identifier; where this was not the case, the information in the ‘Supplemental Material’ and/or ‘Reference’ sections was used. UniProtKB consists of two sections, UniProtKB/SwissProt and UniProtKB/TrEMBL. The former contains manually annotated records with information extracted from literature and curator-evaluated computational analysis, while the latter contains high quality computationally analyzed records enriched with automatic annotation and classification. While mapping to the UniProtKB a UniProtKB/SwissProt entry was preferentially chosen over a UniProtKB/TrEMBL entry. A TrEMBL entry containing the longest version of the sequence was preferentially used where a choice of only TrEMBL entries was available, since the longer entry is most likely to contain the entire protein sequence. In cases where there was no UniProtKB entry and the necessary criteria specified in the Annotation Manual were satisfied a protein entry was made in IntAct database. These had only an EBI accession number. The interactor-pairs were often determined based on the information available in the ‘Figure Legends’ and the ‘Results’ sections of the article.

Protein interaction sentences subtask ISS - ‘source-text’ to describe the interaction:

Multiple techniques may describe the interactions between the same two interactors. These techniques and the interactors they detect are described in various regions of the text of the article. The most pertinent text giving information about the interaction detection method and the protein interactors was stored on the IntAct interaction entry as an annotation using the annotation-topic: ‘source-text’. Either

PDF or HTML forms of the article were used to find the sentences. Many of these protein interaction sentences were taken from the ‘Results’ and ‘Figure Legend’ sections of the article. There was no restriction on the number of sentences forming a single ‘source-text’ description.

Protein interaction method subtask IMS - Mapping of the interaction data to PSI-MI CV:

The information about the experimental technique used to determine interaction was often available in the ‘Materials and Methods’, ‘Figure Legends’, ‘Supplemental Material’ and ‘Results’ sections of the articles. The deepest possible child term of PSI-MI CV root term ‘interaction detection method’ is used to describe the method in a consistent machine-readable form. Where more than one method in an article identified an interaction, the UniProtKB identifiers for the interactors were reported in the context of all the experimental methods used. Hence, the interaction between the same two interactors could be described multiple times.

Assessment of the curation process:

The interaction data entered in the IntAct database by the curators as per the Annotation Manual was checked using an automated procedure based on predefined curation rules and designed to detect common errors. A further manual evaluation was carried out by a senior curator to ensure that the information in the IntAct database correctly represented the information in the publication. The final data representation was as agreed upon between the senior curator and the primary curator. The authors were notified when the IntAct records were released and their examination of the IntAct records provided a third level of quality control.

Release of the test set:

All the articles curated for the BioCreative test set contained the annotation topic ‘dataset’ with a description ‘BioCreative - Critical Assessment of Information Extraction systems in Biology’ on the individual experiment. This allowed organizers to download the entire dataset.

List of abbreviations:

1. Human Proteome Organisation Proteomics Standards Initiative (HUPO-PSI)
2. Controlled Vocabulary (CV)
3. Proteomic standards Initiative - Molecular Interactions (PSI-MI)
4. Journal of Biological Chemistry (JBC)
5. Gene Ontology (GO)

Acknowledgements:

Dr. David Thorneycroft also made substantial amount of contribution to the test and training datasets generated for BioCreative competition.

References:

1. Kerrien S, Alam-Faruque Y, Aranda B, Bancarz I, Bridge A, Derow C, Dimmer E, Feuermann M, Friedrichsen A, Huntley R, Kohler C, Khadake J, Leroy C, Liban A, Lieftink C, Montecchi-Palazzi L, Orchard S, Risso J, Robbe K, Roechert B, Thorneycroft D, Zhang Y, Apweiler R, Hermjakob H: **IntAct – Open Source Resource for Molecular Interaction Data.** *Nucleic Acids Res* 2007, **35**:D561-5.
2. Chatr-aryamontri A, Ceol A, Palazzi LM, Nardelli G, Schneider MV, Castagnoli L, Cesareni G: **MINT: the Molecular INTeraction database.** *Nucleic Acids Res* 2007, **35**:D572-4.
3. Orchard S, Salwinski L, Kerrien S, Montecchi-Palazzi L, Oesterheld M, Stümpflen V, Ceol A, Chatr-aryamontri A, Armstrong J, Woollard P, Salama J, Moore S, Wojcik J, Bader GD, Vida M, Cusick M, Gerstein M, Gavin A, Superti-Furga G, Greenblatt J, Bader J, Uetz P, Tyers M, Legrain P, Fields S, The GO Consortium, Gilson M, Hogue C, Mewes H, Apweiler R, Xenarios I, David Eisenberg2, Cesareni G, Hermjakob H: **The Minimum Information required for reporting a Molecular Interaction Experiment (MIMIx)** *Nature Biotechnology*, in press.
4. The UniProt Consortium: **The Universal Protein Resource (UniProt).** *Nucleic Acids Res.* 2007, **35**:D193-197.

5. Orchard S, Hermjakob H, Binz PA, Hoogland C, Taylor CF, Zhu W, Julian RK Jr, Apweiler R: **Further steps towards data standardisation: the Proteomic Standards Initiative HUPO 3(rd) annual congress, Beijing 25-27(th) October, 2004**; *Proteomics. 2005*, **5**(2):337-9.
6. Orchard S, Montecchi-Palazzi L, Hermjakob H, and Apweiler R: **The Use of Common Ontologies and Controlled Vocabularies to Enable Data Exchange and Deposition for Complex Proteomic Experiments**. *Pacific Symposium on Biocomputing 2006*; **10**:186-196.
7. Cote RG, Jones P, Apweiler R, Hermjakob H: **The Ontology Lookup Service, a lightweight cross-platform tool for controlled vocabulary queries**. *BMC Bioinformatics. 2006*, **28**(7):97.
8. Kerrien S, Orchard S, Montecchi-Palazzi L, Aranda B, Quinn AF, Vinod N, Bader GD, Xenarios I, Wojcik J, Sherman D, Tyers M, Salama JJ, Moore S, Ceol A, Chatr-aryamontri A, Oesterheld M, Stümpflen V, Salwinski L., Neroth J., Cusick ME, Vidal M, Gilson M, Armstrong J, Woppard P, Hogue C, Eisenberg D, Cesareni G, Apweiler R, Hermjakob H: **Broadening the Horizon – Level 2.5 of the HUPO-PSI Format for Molecular Interactions** (In Preparation).
9. Mulder NJ, Apweiler R, Attwood TK, Bairoch A, Bateman A, Binns D, Bork P, Buillard V, Cerutti L, Copley R, Courcelle E, Das U, Daugherty L, Dibley M, Finn R, Fleischmann W, Gough J, Haft D, Hulo N, Hunter S, Kahn D, Kanapin A, Kejariwal A, Labarga A, Langendijk-Genevaux PS, Lonsdale D, Lopez R, Letunic I, Madera M, Maslen J, McAnulla C, McDowall J, Mistry J, Mitchell A, Nikolskaya AN, Orchard S, Orengo C, Petryszak R, Selengut JD, Sigrist CJ, Thomas PD, Valentin F, Wilson D, Wu CH, Yeats C: **New developments in the InterPro database**. *Nucleic Acids Res. 2007*, **35**:D224-8.
10. Gene Ontology Consortium: **The Gene Ontology (GO) project in 2006**. *Nucleic Acids Res. 2006*, **34**:D322-6.
11. BioCreAtIVe II (2006): http://biocreative.sourceforge.net/biocreative_2.html

Figures

Figure1.
An overview of the PSI-MI CV in OLS.

OLS - Ontology Lookup Service

MI Ontology Browser

Associated information

definition	The classical two-hybrid system is a method that uses transcriptional activity as a measure of protein-protein interaction. It relies on the modular nature of many site-specific transcriptional activators (GAL 4), which consist of a DNA-binding domain and a transcriptional activation domain. The DNA-binding domain serves to target the activator to the specific genes that will be expressed, and the activation domain contacts other proteins of the transcriptional machinery to enable transcription to occur. The two-hybrid system is based on the observation that the two domains of the activator need to be non-covalently brought together by the interaction of any two proteins. The application of this system requires the expression of two hybrid. Generally this assay is performed in yeast cell, but it can also be carried out in other organism.
preferred name	two hybrid
related synonym	Gal4 transcription regeneration
related synonym	two-hybrid
related synonym	2h
related synonym	2H
xref_definition	PMID:1946372
xref_definition	PMID:12634794
xref_definition	PMID:10967325

Legend:

- is a
- develops from
- part of
- other

Tables

Table 1

Statistics of the data contributed by IntAct database to the test set.

Test set summary	count
Number of PubMed articles	154
Number of experiments	484
Number of interactions	951
Number of annotation 'source-text' on the interactions	815



IASL Systems in the Gene Mention Tagging Task and Protein Interaction Article Sub-task

Hong-Jie Dai¹

hongjie@iis.sinica.edu.tw

Richard Tzong-Han Tsai^{1*}

thtsai@iis.sinica.edu.tw

Hsi-Chuan Hung¹

yabt@iis.sinica.edu.tw

Wen-Lian Hsu^{1*}

hsu@iis.sinica.edu.tw

¹ Institute of Information Science, Academia Sinica, 115 Nankang, Taipei, Taiwan

1 Overview

Named entity recognition (NER) is a crucial step for information extraction of relationships between genes and gene products. BioCreAtIvE II Gene Mention (GM) tagging task is concerned with this problem. The first part of this paper employs: 1) Conditional random fields (CRF) as the underlying machine learning model, 2) A set of features which are selected by sequential forward search algorithm, 3) Numerical normalization, and 4) Pattern-based post processing to resolve the GM task.

For GM task, we collect training/testing/development dataset from BioCreAtIvE I [1] and II to form a 15,443 sentences training set. In order to make use of this training set, we build a rule-based tokenizer based on the dataset from BioCreAtIvE I Task 1A. This tokenizer is also used to tokenize the training/testing set in our BioCreAtIvE II GM task and Protein Interaction Article Sub-task 1 (IAS).

The second part of this paper is about identifying protein-protein interaction (PPI) related biomedical abstracts. We propose a novel feature representation scheme, contextual-bag-of-words, to exploit named entity information. We further improve the performance by extracting reliable and informative instances from unlabeled and likely positive data to provide additional training data.

This paper is organized as follows. In Section 2 we describe our GM tagging system. In Section 3 we describe our PPI-text classification system. Finally, we conclude our work briefly in Section 4.

2 Gene Mention (GM) Tagging Task

Before describing our system, we first explain the way we used to formulate the NER problem. According to the IOB2 format, we transform the original sentence into a token/tag format. For example, the sentence “Comparison with alkaline phosphatases and 5-nucleotidase” will be transformed to “Comparison/O with/O alkaline/B phosphatases/I and/O 5-nucleotidase/B”.

2.1 System Description

After formulating the NER problem, we use seven feature types, including word, bracket, orthographical, part-of-speech (POS), affix, character-*n*-gram, and lexicon, to represent the characteristics of biomedical name entities (NEs). We explain them in the next section.

In order to leverage the performance and memory usage, we employ sequential forward selection (SFS) algorithm to find the best feature set and numerical normalization to reduce the number of features. Finally, we apply global patterns to fix the tag dependency outside the context window.

2.1.1 Feature Selection

It is inefficient to include all features in a Bio-NER model since memory resources are limited, and some features are ineffective. For our dataset, we divide it into a training set (10,298 sentences) and a development

* corresponding authors

set (5,153 sentences). Due to time and space limitations, it is very difficult to select a globally optimal feature set for the development set. We employ sequential forward selection algorithm to find the best feature set.

The algorithm is described as follows. We first calculate which feature has the highest F-score and select this feature as the basis for the feature pool. In each subsequent iteration, we individually add one feature type to the feature pool and calculate their F-scores, each time selecting the best scoring feature type and adding it to the pool. This process continues until the F-score stops increasing.

2.1.2 Numerical Normalization

In addition to selecting the efficient feature set that maximizes performance with limited memory resources, we also apply numerical normalization to reduce the number of features in each feature set. According to our observation, some proteins or genes of the same family usually differ in their numerical parts. For example, interleukin-2 and interleukin-3 belong to the same family—interleukin. In Bio-NER, they are both the target NE. Therefore, we normalize all numerals into one. For example, both interleukin-2 and interleukin-3 are normalized to interleukin-1.

2.1.3 Using Global Pattern to Improve CRF

The sequential tagging models we applied usually follows the Markov assumption that the current tag only depends on the previous tag. However, in Bio-NER, there are many exceptions. An NE may depend on the previous or next NE, or words among these NEs. Common sequential models cannot model this dependency. Furthermore, the sequential model only uses the information in the limited context window. It may fail if there are dependencies beyond the context window. To alleviate these problems, we apply global patterns composed of NEs and surrounding words.

Global Pattern Induction and Filtering

The first step in creating global patterns is to apply numerical normalization to all sentences in the training, development, and test sets. For each pair of sentences in the training set, we apply the Smith-Waterman local alignment algorithm [2] to find the longest common string, which is then added to the candidate pattern pool. During the alignment process, for each position, either of the two inputs that share the same word or NE can be counted as a match. The similarity function used in the Smith-Waterman algorithm is:

$$\text{Sim}(x, y) = \max \begin{cases} 1, & x = y \\ 1, & x's \text{ tag is } B \text{ or } I \text{ and } y's \text{ tag is } B \text{ or } I \\ 0, & \text{otherwise} \end{cases}$$

where x and y referred to any two compared tokens from the first and second input sentences, respectively. The similarity of two inputs is calculated by the Smith-Waterman algorithm based on this token-level similarity function.

Then we illustrate how patterns are extracted from a sentence pair in the training set. Given the following two tagged sentences:

...chemical/O interactions/O that/O **inhibit**/O butyrylcholinesterase/**B and**/O ...

and

...combinations/O of/O chemicals/O that/O **inhibit**/O butyrylcholinesterase/**B and**/O ...

, we will generate the "**inhibit <NE> and**" pattern. Here, we use bold face for the aligned words and tags in bold font. The first and last tokens in a pattern are constrained to be words, sentence beginning or ending symbols.

The extracted patterns are composed of a headword, NE type and a tail-word, e.g., "headword <NE type> tail-word." To test its effectiveness, each pattern is applied to the development set to correct the NE tags of all sentences. If the pattern's error ratio exceeds a certain threshold, τ , it is filtered out.

2.2 Feature Set

2.2.1 Word and bracket Features

Words preceding or following the target word may be useful for determining whether it is an NE or not. We use window size from -1 to 1, that is, the previous word, current word, and next word. We also include a feature to indicate whether the current token occurs within brackets or inside quotations.

2.2.2 Character-*n*-gram Features

A character *n*-gram is a substring of *n* characters of a longer string [3]. This feature helps our system to recognize NEs according to certain informative substrings, such as "ase" in "decarboxylase". In our system, we use character substrings of length 3 to 4 characters.

2.2.3 Orthographical Features

Table 1 lists all orthographical features used in our system. These features are widely used in other general NER [4] or biomedical NER systems [5].

Table 1: Orthographical features

Feature name	Regular Expression
INITCAP	^[A-Z].+
CAPWORD	^[A-Z][a-z]+\$
ALLCAPS	^[A-Z]+\$
CAPSMIX	^[A-z]*([A-Z][a-z][a-z][A-Z])[A-z]*\$
ALPHANUMMIX	^[A-z0-9]*([0-9][A-z][A-z][0-9])[A-z0-9]*\$
ALPHANUM	^[A-z]+[0-9]+\$
UPPERCHAR	^[A-Z]\$
LOWERCHAR	^[b-z]\$
SHORTNUM	^[0-9][0-9]?\$
INTEGER	^-?[0-9]+\$
REAL	^-?[0-9]\.[0-9]+\$
ROMAN	^[IVX]+\$
HASDASH	-
INITDASH	^_
ENDDASH	-\\$
PUNCTUATION	^[.,;?!]\$
QUOTE	^["'"]\$

2.2.4 POS Features

POS information is quite useful for identifying named entities. The GENIA POS tagger [6] and MEDPOST tagger [7] are used to provide POS information.

2.2.5 Affix Features

Affixes including prefixes and suffixes are morphemes. They are attached to base morphemes, such as roots, or to stems, to form words. Some of them can provide information to identify NE. For example, words ending in "~ase" are usually proteins. The length we used for prefixes and suffixes is 2-4 characters.

2.2.6 Lexicon Features

Finally, we include two kinds of lexicon features: exact match and dictionary distance. The first kind is just a binary feature indicating whether a token occurs in our lexicon or not.

In reality, it is difficult to find a lexicon which contains all possible variations of biomedical names. Therefore, it is useful to measure the distance between tokens and words in an external lexicon and set this as a feature. We use the Jaro-Winkler distance metric to compute the minimum distance between a token *x* and

an entity e in lexicon. These features are useful [8] because partial matches to entity names are informative. The lexicons we used are extracted from HUGO [9] and BioCreAtIvE I dataset.

2.3 Results

Table 2 shows the result of our three runs in BioCreAtIvE II test set. The best F-Measure is Run 3 which uses all seven feature types and applies post processing. We can see that adding lexicon features increases the precision of our system by 0.13%.

Table 2: Final results

Run ID	Run	Precision	Recall	F-Measure
1	No-lexicon feature	92.69%	68.73%	78.93%
2	With lexicon feature	92.82%	68.82%	79.04%
3	Post processing	92.67%	68.91%	79.05%

Table 3 shows the results of our system on the development set, which are relatively balanced in precisions and recalls in the development set. However, in the test set, our system achieves higher precisions but lower recalls. We believe that this is due to the strategy we used to create gold standard for the development set. Our development set is selected from training sets in BioCreAtIvE I and II. Some selected sentences exist in both BioCreAtIvE I and II datasets. These sentences are sometimes tagged differently in BioCreAtIvE I and II. We treat the BioCreAtIvE II annotation as the gold standard and BioCreAtIvE I as the alternative answers. Therefore, there may be many alternative answers for an NE in the development set. But in BioCreAtIvE II's test set, the gold standard was not created in this way. We believe that on average, the number of alternative answers per NE in the test set is less than that in the development set. This phenomenon causes the lower recalls in the test set.

Table 3: The performance on our development set

Run	Precision	Recall	F-Measure
No-lexicon feature	78.40%	81.75%	80.04%
With lexicon feature	78.86%	81.51%	80.17%

3 Protein Interaction Article Sub-task (IAS)

Before extracting PPI information from biomedical abstracts, it is necessary to identify them in the ever-increasing corpus of biomedical abstracts. This is the purpose of the BioCreAtIvE II IAS task. This task can be formulated as a text classification (TC) problem in the biomedical domain. We consider the following three critical issues in developing our PPI-TC system.

Adopting Contextual Information. In TC, documents are usually represented by bag-of-words (BoW) features. However, in PPI-TC, some words are informative only in certain contexts. For example, "bind" is more informative in indicating if an abstract is PPI-relevant when it appears in a sentence that has at least two proteins.

Filtering Out Likely Positive Instances. Gene Ontology (GO) is a widely used taxonomy that classifies many discovered protein interaction types, whereas a PPI database usually contains only some specific types that may not satisfy our requirements. Therefore, we usually treat abstracts annotated in PPI databases as likely positive (LP) examples. Those abstracts that do not contain PPI types of interest need to be filtered out.

Selecting Likely Negative Instances. It is easy to acquire a large number of positive (PPI-relevant) abstracts from PPI databases for use as LP data. On the other hand, likely-negative (LN) instances are often quite scarce. Since, most machine learning (ML) models used in classification require a balanced number of LP and LN examples, we must select more LN instances.

3.1 Method

3.1.1 Support Vector Machines and Term Weighting

The support vector machine (SVM) model is one of the best known ML models that can handle sparse high dimension data, which has been proved useful for text classification [10]. It tries to find a maximal-margin separating hyperplane $\langle \mathbf{w}, \varphi(\mathbf{x}) \rangle - b = 0$ to separate the training instances, i.e.,

$$\min \|\mathbf{w}\|^2 + C \sum_i \xi^{(i)} \quad \text{subject to } y^{(i)}(\langle \mathbf{w}, \varphi(\mathbf{x}^{(i)}) \rangle - b) \geq 1 - \xi^{(i)}, \quad \forall i$$

where $\mathbf{x}^{(i)}$ is the i th training instance which is mapped into a high-dimension space by $\phi(\cdot)$, $y_i \in \{1, -1\}$ is its label, $\xi^{(i)}$ denotes its training error, and C is the cost factor (penalty of the misclassified data). The mapping function $\phi(\cdot)$ and the cost factor C are the main parameters of a SVM model.

When classifying an instance \mathbf{x} , the decision function $f(\mathbf{x})$ indicates that \mathbf{x} is "above" or "below" the hyperplane. [11] shows that the $f(\mathbf{x})$ can be converted into an equivalent dual form which can be more easily computed:

primal form: $f(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \varphi(\mathbf{x}) \rangle - b)$; dual form: $f(\mathbf{x}) = \text{sign}\left(\sum_i \alpha^{(i)} y^{(i)} K(\mathbf{x}^{(i)}, \mathbf{x}) - b\right)$

where $K(\mathbf{x}^{(i)}, \mathbf{x}) = \langle \varphi(\mathbf{x}^{(i)}), \varphi(\mathbf{x}) \rangle$ is the kernel function and $\alpha^{(i)}$ can be thought of as w 's transformation.

In the IAS subtask, we chose the following polynomial kernel according to our preliminary experiment results:

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = (\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle + 1)^2 \text{ and } C = 1$$

In the text categorization problem, a document d is usually represented as a term vector \mathbf{v} . Each dimension v_i in \mathbf{v} corresponds to a term t_i . v_i is calculated by a term weighting function, which is very important to SVM-based text categorization because SVM models are sensitive to the data scale, namely dominated by some widest dimensions. In this paper, we employ the three most popular functions: Binary, TFIDF, and BM25, which are defined as follows:

$$\text{Binary}(t_i, d) = \begin{cases} 1 & \text{if } t_i \text{ appears in } d \\ 0 & \text{otherwise} \end{cases},$$

$$\text{TFIDF}(t_i, d) = \text{TF}(t_i, d) \cdot \text{IDF}(t_i, D)$$

where D is the document set that contains all documents in the training and test sets,

$$\text{TF}(t_i, d) = \frac{t_i \text{'s frequency in } d}{\text{word counts of } d}, \text{ and } \text{IDF}(t_i, D) = \frac{\#\text{documents} \in D \text{ containing } t_i}{|D|}$$

BM25's definition of can be found in [12].

3.1.2 Methods of Exploiting Named Entity Information

A PPI abstract must contain some protein names. Hence, recognition of protein names in abstracts can improve the identification of PPI abstracts. We use our GM tagging system to provide NEs information. In the following we describe our new feature representation scheme.

Contextual Bag of Words (CBoW). The number of protein names that exist in the context affects a word's informativeness for PPI relevance. Based on this fact, we distinguish the original word bags into different contextual bags. The words in individual sentences are bagged according to the number of protein named entities (NEs) in the sentence. If there are 0 NEs the words are put into contextual bag 0; if 1 NE, then bag 1; and if 2 or more NEs, then bag 2.

For comparison, we implement two well-known features that should be incorporated with BoW features:

Bag of Phrases (BoP). [13] suggested that adding phrases into the original bags can retain some order information which is lost in BoW. In our case, we add protein NE phrases into bags.

Bag of Normalized NEs (BoN). The more protein names that appear in an abstract, the more likely it is to be PPI-relevant. Following [14], we replace each NE in a given abstract with “PROTEIN_i”, where *i* denotes the order of appearance in this abstract. Abstracts containing different numbers of NEs have different normalized NE features.

3.1.3 Filtering Out Likely-Positive Instances and Selecting Likely-Negative Instances

To filter out irrelevant data from likely-positive data, we use the initial model that is trained on TP+TN using only BoW features. Those abstracts in the original LP with an SVM output in $[\gamma^+, 1]$ are retained, where γ^+ is chosen to be 0. The dataset produced by filtering out irrelevant LPs is referred to as selected likely positive data (LP*).

To select likely negative instances, we employ a bootstrapping-like technique inspired by [15]. We collect 50k unlabeled abstracts from the PubMed biomedical literature database and classify them with our initial model. The articles with an SVM output in $[-1, \gamma^-]$ form the selected likely-negative (LN*) dataset, where $-1 < \gamma^- < 0$ is a threshold. γ^- is chosen to be -0.9. The articles with predicted values less than -1 are excluded since they are absolutely negative examples that may not be useful for determining the hyperplane in SVM. In addition, the instances whose SVM outputs are in $[\gamma^-, 0]$ are discarded due to unreliability.

3.2 Results

Three datasets provided by BioCreAtIvE II are shown in Table 4. For each abstract, we remove all punctuation symbols, numbers, and stop words in the preprocessing step. We use our GM tagging system to tag NEs in each abstract. Before applying our system to the test set from BioCreAtIvE II IAS task, we conduct 10-fold cross validation experiments on the training set and use the F-Measures to score our system.

Table 4: Three datasets in IAS

Dataset	Size
True positive (TP)	3536 abstracts
True negative (TN)	1959 abstracts
Likely-positive (LP)	18930 abstracts

3.2.1 Exploiting Named Entity Information

Table 5 shows the 10-fold cross validation results on the training set for different IAS methods that exploit NE information. CBoW appears to outperform BoW, whereas the other two configurations that incorporate NE features into BoW only slightly improve the performance of BoW regardless of the weighting.

Table 5: F-Measures of different IAS methods of using NEs

Features	binary	TF-IDF	BM25
BoW	93.85	94.04	94.41
BoW + BoP	94.01	94.15	94.47
BoW + BoN	94.71	94.92	94.70
CBoW	95.85	96.01	97.34

3.2.2 Expanding the Training Set

In this section, we examine the effects of adding LP* and LN*. Without loss of generality, we use the CBoW representation scheme. As shown in Table 6, adding the selected data slightly improves the F-Measure of all weight schemes.

Table 6: F-Measures of original training set vs. the expanded one

Configuration	binary	TF-IDF	BM25
TN+TP	95.85	96.01	97.34
TN+TP+LN*+LP*	96.16	96.18	97.91

3.2.3 Results of IAS Task

Table 7 shows the results on the test set, including our IAS system’s performance along with the mean and

median scores of all the participant systems. Our Run1 system employs the best feature set found in the development set. It uses the LP* and LN* data while our Run2 system does not. We can see that with LP* and LN*, the performance can be slightly improved by 1.10%. These results are similar to those in Table 6. In addition, both Run 1 and 2 significantly outperform the mean and median scores. This shows that our CBoW representation is generally effective in the IAS task.

Table 7: Performance on the test set

Configuration	Precision	Recall	F-Measure
Run 1 (TN+TP+LN*+LP*)	68.90%	85.07%	76.13%
Run 2 (TN+TP)	66.46%	86.13%	75.03%
Mean	66.42%	76.36%	68.68%
Median	67.72%	85.07%	72.24%

4 Conclusions

In the work, we first propose a NER system in the biomedical domain using SFS feature selection and numerical normalization to efficiently utilize the memory and maximize tagging performance. We use the Smith-Waterman local alignment algorithm to help ML-based Bio-NER to deal with extremely difficult cases which need longer context windows.

Finally, we propose a novel CBoW feature representation scheme and demonstrate that it outperforms other methods that also exploit NE information in PPI-TC. We also extract likely positive and likely negative data for enhancing the performance of PPI-TC. Our study of the PPI-TC problem presents a potential new direction of exploiting NLP-based contextual information.

References

- [1] Hirschman, L., Yeh, A., Blaschke, C., and Valencia, A., Overview of BioCreAtIVe: critical assessment of information extraction for biology, *BMC Bioinformatics*, 2005.
 - [2] Smith, T.F. and Waterman, M.S., Identification of common molecular subsequences, *J. Mol. Biol.*, 147(1):195-197, 1981.
 - [3] Cavnar, W.B., Using an n -gram-based document representation with a vector processing retrieval model, *Proc. TREC-3 Conference*, 269–278, 1994.
 - [4] Florian, R., Ittycheriah, A., Jing, H., and Zhang, T., Named entity recognition through classifier combination, *Proc. CoNLL-03 Conference*, 168-171, 2003.
 - [5] Zhou, G. and Su, J., Exploring deep knowledge resources in biomedical name recognition. *Proc. JNLPBA-04 Conference*, 2004.
 - [6] Tsuruoka, Y. and Jun'ichi Tsujii, J., Bidirectional inference with the easiest-first strategy for tagging sequence data. *Proc. HLT/EMNLP-05 Conference*, 2005.
 - [7] Smith, L., Rindflesch, T., and Wilbur, W.J., MedPost: a part-of-speech tagger for bioMedical text. *Proc. Bioinformatics*, 2004.
 - [8] Cohen, W.W. and Sarawagi, S., Semi-markov conditional random fields for information extraction, *Proc. NIPS-04 Conference*, 2004.
 - [9] <http://www.gene.ucl.ac.uk/nomenclature/>
 - [10] Joachims, T., Text categorization with support vector machines: learning with many relevant features. *Proc. ECML-98. Conference*, 1998.
 - [11] Cristianini, N. and Shawe-Taylor, J., *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.

- [12] Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M., Okapi at TREC-3, *Proc. TREC-3 Conference*, 1994.
- [13] Scott, S. and Matwin, S., Feature engineering for text classification, *Proc. of ICML-99 Conference*, 1999.
- [14] Paradis, F. and Nie, J.-Y., Filtering contents with bigrams and named entities to improve text classification, *Proc. of AIRS-05 Symp.*, 2005.
- [15] Liu, B., Lee, W.-S., Yu, P., and Li, X., Partially supervised classification of text documents, *Proc. ICML-02 Conference*, 2002.



Identifying Gene Mentions by Case-Based Classification

Mariana Lara Neves¹
marianaln@hotmail.com

¹ PhD Student, Facultad de Informática, Universidad Complutense de Madrid, Madrid, Spain

Abstract

The work presented here proposes a case-based classification for the gene mention task in the BioCreAtIvE 2 challenge. The classification performed by the system for each word in an article is based on the selection of the best or more similar case in a base of known and unknown cases. The procedure showed good results, precision of 71.68 and recall of 62.33.

Keywords: text mining, gene mention, case-based reasoning.

1 Introduction

The work presented here reports the submission to the Gene Mention task of the BioCreAtIvE 2 competition. The method proposed is of the identification of gene mentions in a text document by a binary classification of the words.

The classifier created uses the case-based reasoning [1] foundations in which in a first step several cases of all of the classes involved in the problem are stored in a base to be further used in the classification of a new case. The system must search the base for the case most similar to the problem and the classification decision is given by the class of the case selected as the most similar.

2 Method and Results

2.1 Construction of Case Bases

The documents provided for the Gene Mention task were divided in 10 subsets so as to perform a 10-fold cross validation training. For each of the sets, documents had their sentences and words extracted, the words present in a stopwords list [3] were removed and the remaining ones were used to construct the two case bases, for the known and unknown cases. The procedure presented here is similar to the one proposed for the part-of-speech tagging task in [2].

The known case base is composed of words that are present in the training documents and its function is to classify these known words when they appear in new document. For the best training set of the cross-validation runs, 40298 was the number of cases acquired. Some examples are shown in Figure 1. In the figure, the gene mention cases are highlighted with a symbol and its attributes are presented according to its four attributes described below:

1. W: word that is present in the training document, no matter if it is a gene mention or not;
2. G: boolean indicative of it being or not a gene mention;
3. B: boolean indicative of the preceding word being a gene mention or not;
4. #: frequency of the case: as each case is unique, this attribute correspond to the number of times that the three other attributes appeared with the same values in the whole training set.

The unknown case base is composed by the format of the words that are present in the training documents, not of the words themselves as its function is to classify words unknown to system that may appear in a new document. For the best training set of the cross-validation, 3270 was the number of unknown cases acquired. Some examples are shown in Figure 2. In the figure, gene cases are highlighted with a symbol, an example of the original word is shown and its attributes are presented according to the attributes described below:

1. F: format of the word that is present in the text, no matter if it is a gene mention or not, according to the code of letters following described;
2. G: boolean indicative of it being or not a gene mention;
3. B: boolean indicative of the preceding word being a gene mention or not;
4. #: frequency of the case: as each case is unique, this attribute correspond to the number of times that the three other attributes appeared with the same values in the whole training set.

As for the first attribute of format, each word was converted to a sequence of codes (letters) according to its characteristics. Complete words or parts of words that are present in a biological lexicon ("protein", "gene", "promoter") are substituted for the code "W", Greek letters ("alpha", "gamma") for "G", special suffixes ("ase", "ine") for "S", upper cases for "M", numbers for "N", lower case letters for "L" and the remaining symbols are kept in its original format.

W = cd8 G = true B = false # 2	W = pep1 G = true B = true # 1	W = xylene G = false B = false	W = amycolata G = false B = true
---	---	---	---

Figure 1: Examples of known cases.

F = MM-N-LS Ex. AP-1-luciferase G = true B = false # 1	F = G/MM Ex. alpha/CG G = true B = true # 1	F = GN-N-L Ex. alpha1-2-linked G = false B = false	F = NMN Ex. 3T3 G = false B = true
--	---	---	--

Figure 2: Examples of unknown cases.

2.2 Classification of the words

As for the classification step, new documents are processed similar to the training ones. Sentences and their words are extracted, stopwords are removed and the remaining words are the ones to be classified by the system as if they are a gene mention or not.

For each of these words, the system first verifies of its presence in the known case base, as to check if the word appeared in the training documents. The system initially looks for a case in which the word is present but also the boolean indicative of the preceding word (attribute B) is the same, so as to select the most similar case to the actual situation. Two are the possibilities of cases to be found, one with the gene indicative (attribute G) set to true and another one to false. If these two cases are found, the boolean indicative of the case with higher frequency is selected and this indicative is actually the final classification answer of the system to the word. If just one of them is found, this is the one used for the classification. If an exact case is not found (same word, same boolean indicative of the preceding word), the system looks for a case with the opposite indicative of the preceding word. Similarly, two are the possibilities to be found and the one with the higher frequency will be selected by the system. If the word in consideration is the first word of the sentence, the attribute boolean indicative of the preceding word is considered as "false", otherwise the system sets for this attribute the value obtained in the classification of the preceding word.

If a word cannot be found in the known case base, a search in the unknown case base is performed. The word is then converted to the sequence of codes that represent its format as this is main attribute of the case to be searched in the base. Similarly to the known case base, the system first looks for a case with the same format

and same boolean indicative of the preceding word and of the two possible cases found, the one with the higher frequency is selected. Otherwise, a search with the opposite indicative of the preceding word is performed and the case with higher frequency is selected. If no case is found in the unknown case base, it means that the format of the word is a new one, not present in the training documents, and so the word is tagged as a gene mention as default.

2.3 Results

The output file required by BioCreAtIvE 2 challenge should be composed of the gene mentions start and end position in the original document. As the system classifies each word according to it being or not a gene mention, the final list was created by considering the sequences of gene mention words as being a unique mention.

As for the best results obtained by the system, Table 1 shows the precision, recall and F-measure of the training and the test sets. The results are encouraging as the procedure used in this work is relatively simple.

Table 1: Results of the system.

Table 1: Results of the system			
Corpus	Precision	Recall	F-Measure
Training/Development	76.36	62.20	68.56
Test	71.68	62.33	66.68

3 Discussions

Many of the false negative problems were due to multi-word gene mentions in which the words of its composition usually appeared as non gene mention in the case bases but when used together in a text article corresponded to a true gene mention. The cases used in this work were all of a single word and the consideration of multi-word cases is a possibility to overcome this problem.

The representation of the format must also be reviewed as many false negative and false positive errors were found especially for words composed of a mix of upper cases, lower cases and numbers. The system repeats the upper case code ("M") when found more than once (see examples of Figure 2) but not for the lower case code ("L") and the number code ("N") in the same situation.

As for the false positive problems, the system was not able to find many multi-word gene mentions, returning just parts of it in the output file. The consequence is that a unique gene mention affects twice the results of the system, as a wrong positive error and a missing negative one. Many false positive errors were also due to words that could not be found in any of the bases and that were classified as positive (gene mention) as default.

References

- [1] Aaamod A., and Plaza E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *Artificial Intelligence Communications*, vol. 7, Nr. 1, pages 39-59, 1994.
 - [2] Daelemans, W., Zavrel, J., Berck, P., and Gillis, S., (1996) MBT: A Memory-Based Part of Speech Tagger-Generator. *Fourth Workshop on Very Large Corpora*, pages 14-27, Copenhagen, Denmark, 1996.
 - [3] <http://www.unine.ch/info/clef/englishST.txt>



Combined Conditional Random Fields and n -Gram Language Models for Gene Mention Recognition

Craig A. Struble¹

craig.struble@marquette.edu

Richard J. Povinelli²

richard.povinelli@marquette.edu

Michael T. Johnson²

mike.johnson@marquette.edu

Dina Berchanskiy¹

dina.berchanskiy@marquette.edu

Jidong Tao²

jidong.tao@marquette.edu

Marek Trawicki²

marek.trawicki@marquette.edu

¹ Department of Mathematics, Statistics and Computer Science, P.O. 1881, Milwaukee, WI 53201-1881, USA

² Department of Electrical and Computer Engineering, P.O. 1881, Milwaukee, WI 53201-1881, USA

Abstract

In this paper, we propose the use of character n -gram and multiple conditional random field (CRF) models for BioCreAtIvE 2 Task 1, gene/protein name recognition. We investigated different state transition weighting schemes for CRFs and discovered that models provided independent non-overlapping mentions. To improve recall, the results of multiple models are combined. To improve precision, character n -gram models classify gene/protein mention containing sentences. Our best approach achieved a precision of 84.35%, recall of 81.39% and F-measure of 82.85%.

Keywords: conditional random field, named entity recognition, n -gram models

1 Introduction

Effective automated tools for identifying gene mentions can help in rapidly creating large gene centric knowledge bases, identifying associations between genes and diseases, and indexing biomedical literature by genes and their products. In 2006, the BioCreAtIvE 2 community challenge provided training, development and evaluation data to critically assess information extraction techniques for several text processing tasks motivated by the biological community [2, 3].

In this paper, we present a method for identifying gene/protein mentions using multiple conditional random field (CRF) [4] and n -gram language models. Our system is similar to McDonald and Pereira's CRF-based tagger in the first BioCreAtIvE contest [6], but utilizes different features and combines multiple models. Other CRF-based tagging systems for biological named entity recognition include ABNER [8] and GeneTaggerCRF [9]. Systems primarily differ in their choice of features, CRF parameters and training data, while achieving similar performance.

The system is described in more detail in Section 2. Evaluation of the system and a brief discussion is in Section 3.

2 System Description

Our system treats the problem of identifying gene/protein names as one of tagging a sequence of tokens with labels indicating the location of gene/protein mentions. Sentences are tokenized into numbers with optional decimals and leading + or -, alphanumeric strings with single quotes (to create tokens such as 5'), and individual punctuation marks. For training and tagging, tokens are labeled with one

of three labels *B-GENE*, *I-GENE*, and *O* representing the beginning, inside and outside of a gene mention.

Conditional Random Fields Gene mention tagging employs linear-chain conditional random fields (CRFs), a conditional probability model for tagging sequences [4]. The conditional probability $P(\mathbf{s}|\mathbf{o})$ of a state sequence $\mathbf{s} = s_1, \dots, s_n$ corresponding to labels given the observed token sequence $\mathbf{o} = o_1, \dots, o_n$ is defined by

$$P(\mathbf{s}|\mathbf{o}) = \frac{1}{Z(\mathbf{o})} \exp \left(\sum_{i=1}^n \sum_{j=1}^m \lambda_j f_j(\mathbf{s}, \mathbf{o}, i) \right),$$

where $Z(\mathbf{o})$ is a normalization factor over all state sequences, $f_j(\mathbf{s}, \mathbf{o}, i)$ is a feature function and λ_j is a learned feature weight. The feature functions are written in their most general form.

We developed two CRF models with different Markov-order structures. One is a second-order structure, evaluating the feature function using the current and previous states. Feature functions are represented by $f_j(s_{i-1}, s_i, \mathbf{o}, i)$. The second is a first-order structure, evaluating feature functions in the context of only the current state. Feature functions are represented by $f_j(s_i, \mathbf{o}, i)$. This second model is also known as a *half label* model in the Mallet library [5].

Combining CRF models When evaluating the two CRF models, we noted that performance was similar but the models identified independent non-overlapping gene name mentions. This observation led us to combine the two CRF models using a simple approach in the hopes of improving recall without impacting precision too much. To combine models, one CRF model is chosen as the baseline tagger. The second model is used to assign gene mentions that do not overlap at all with the baseline tagger.

Character n -gram Models In some cases, sentences not containing mentions were tagged. This typically happens when orthographic features of a token strongly indicate that the token is part of a gene mention (e.g., all capital letters). To improve precision, a 6-gram character language model predicted whether or not a sentence contains a gene mention. The n -gram classifier uses untokenized sentences as input. When the n -gram model is used, only sentences predicted to contain mentions are tagged by CRF models.

Features We utilized boolean features of the text being labeled. Orthographic features were used including: the token itself, all capital letters, all lowercase letters, punctuation, quote, alphanumeric, lowercase letters followed by capital letters, initial capital letter, single capital letter, single letter, all alphabetic, single digit, double digits, integer, real number, contains a digit, three letter amino acid code, contains *globin* or *globulin*, contains a Roman numeral, or contains a Greek letter. Additional features included all prefixes and suffixes of lengths 2–4 and whether a token is part of a short form or long form of an abbreviation definition [7]. Contextual features included all features of the 2 preceding and 2 following tokens.

Post Processing A simple post-processing step was used to ignore gene mentions that contained mis-matched parentheses, which indicated a tagging mistake

Implementation The system was implemented in Java using the Mallet [5] and LingPipe [1] libraries.

3 Results and Discussion

During development, the provided set of 15,000 sentences was split into a training set and test set containing 10,500 and 4,500 sentences respectively. For the final submission, all 15,000 sentences were used for training and testing was performed on a blind collection of 5,000 sentences. Precision, recall,

Table 1: System performance on test data. Quartile placement is shown in parentheses.

Submission	Precision	Recall	F-Measure
Combined CRFs without <i>n</i> -gram	84.35 (3)	81.39 (2)	82.85 (2)
Combined CRFs and <i>n</i> -gram	87.53 (1)	77.52 (3)	82.22 (2)
Second-order and <i>n</i> -gram	88.88 (1)	76.02 (3)	81.95 (2)

F-Measure and quartiles for each submission are in Table 1. The results are comparable to McDonald and Pereira [6], with slight improvements in recall and F-measure. As hoped, the combined CRFs improve recall without impacting precision too much. The *n*-gram models improve precision and may be desirable in situations where mislabeling is problematic.

Two classes of gene mentions were problematic. The first was due to gene mention coordination, such as in *clotting factors II, V, VIII, IX, X*. Often only the first part, *clotting factors II*, was tagged resulting in a false positive and false negative contributions. The second was due to parenthesized tokens embedded in the mention, such as in *serum neutralizing (SN) antibody*. Often, the first part, *serum neutralizing*, the part preceding the closing parenthesis, *serum neutralizing (SN*, or the part following the opening parenthesis *SN) antibody*, was tagged. Apparently, clear cues for the proper tagging of parentheses, which are included sometimes, are not learned.

In summary, we obtained modest improvements in recall and F-measure by combining multiple CRFs. Recall and precision could be improved by investing more effort in handling coordination and mentions with embedded parenthesized terms.

References

- [1] alias i. LingPipe. <http://www.alias-i.com/lingpipe/index.html>, 2006. Version 2.3.0.
- [2] BioCreAtIvE II: Critical assessment for information extraction in biology challenge (2006–2007). http://biocreative.sourceforge.net/biocreative_2.html.
- [3] L. Hirschman, A. Yeh, C. Blaschke, and A. Valencia. Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6 Suppl 1:S1, 2005.
- [4] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
- [5] A. K. McCallum. MALLET: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, 2002.
- [6] R. McDonald and F. Pereira. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, 6 Suppl 1:S6, 2005.
- [7] A. S. Schwartz and M. A. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. *Pac Symp Biocomput*, pages 451–462, 2003.
- [8] B. Settles. Biomedical Named Entity Recognition using Conditional Random Fields and Rich Feature Sets. In *COLING 2004 International Joint workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP) 2004*, pages 107–110, 2004.
- [9] R. Talreja. GeneTaggerCRF. http://www.cis.upenn.edu/datamining/software_dist/biosfier/.



Tackling the BioCreative2 Gene Mention task with Conditional Random Fields and Syntactic Parsing

Andreas Vlachos¹ av308@cl.cam.ac.uk

¹ Computer Laboratory, University of Cambridge, CB3 0FD, Cambridge, UK

Abstract

This paper presents an approach to Gene Mention tagging using Conditional Random Fields (CRFs) and syntactic parsing, by taking advantage of the flexibility of the former in order to add features from the output of the latter. We did not use any material or information other than the training data provided in order to maintain the domain independence of the system. Nevertheless, the resulting system achieved 82.84% F-score, which places it in the second performance quartile of the competition.

Keywords: CRFs, syntactic parsing, gene mention tagging

1 Introduction

In this paper we describe our participation in the BioCreative Gene Mention tagging task. The main components used were the Conditional Random Fields implementation (CRFs) [2] from MALLET [3] and the RASP tokenizer, part-of-speech (POS) tagger, lemmatizer and syntactic parser [1]. CRFs were chosen due to the recent success in similar named entity recognition (NER) tasks [4], as well as their flexibility in adding features. The latter aspect we intend to take advantage of in our system, by adding linguistic features from the output of the various components of the RASP toolkit. No other resources were used, therefore the system presented here could be used for other NER tasks. Our expectation is that the combination of deep linguistic analysis and a state-of-the-art statistical model should be able to achieve competitive performance without using domain-specific resources.

2 Methods

As a first step we created tokenized training data from the materials provided, which were a list of sentences with two sets of annotations. We used only the first set of annotations (from the GENE.eval file) in order to annotate the sentences. Then we tokenized the text using RASP's domain independent tokenizer, adding as token boundaries the gene mention boundaries from the annotations. We used the BIEWO scheme for labelling the resulting tokens – the first token of a multitoken mention is tagged as B, the last token as E, the inner ones as I, single token mentions as W and tokens outside an entity as O. In our experiments we found that we obtained better performance with this scheme than with the standard IOB format, possibly due to the large number of multi-token gene mentions and their overlap with common English words or biomedical terms. For each token we extracted the simple orthographic features listed in Table 1.

Then we pass each tokenized sentence to RASP's syntactic parser. We parameterized RASP to pass multiple POS tags per token to the parser to ameliorate unknown word errors and used the grammatical relations (GRs) output from the top-ranked parse. The output of RASP (without the XML tags for brevity) looks like this:

Table 1: Simple orthographic features

the token itself	if it contains digit(s)
if it is alphanumeric	if it contains only digits
if it is alphabetic	if it contains dash(es)
if it is titlecase	if it contains dot(s)
if it is lowercase	if it contains any punctuation marks
if it is uppercase	if it contains punctuation marks and digits
if it is mixed case	2 and 3 letter prefixes and suffixes

```
("No" "post-operative" "haemorrhages" "from" "the"
"prostheses" "were" "observed" ".")
```

```
(|ncsubj| |observe+ed:8_VVN| |haemorrhage+s:3_NN2| _)
(|aux| |observe+ed:8_VVN| |be+ed:7_VBDR|)
(|passive| |observe+ed:8_VVN|)
(|det| |haemorrhage+s:3_NN2| |No:1_AT|)
(|ncmod| _ |haemorrhage+s:3_NN2| |from:4_II|)
(|dobj| |from:4_II| |prosthesis+s:6_NN2|)
(|det| |prosthesis+s:6_NN2| |the:5_AT|)
(|ncmod| _ |haemorrhage+s:3_NN2| |post-operative:2_JJ|)
```

The features extracted from RASP's output for each token are listed in Table 2. It must be noted at this point that the features added from the output of RASP may contain noise, since syntactic parsing is a very complicated task.

Table 2: Features extracted from the output of RASP

the lemma and the POS tag(s) associated with the token
the lemmas for the previous two and the following two tokens
the lemmas of the verbs to which this token is subject (<i>ncsubj</i> relation)
the lemmas of the verbs to which this token is object (<i>dobj</i> relation)
the lemmas of the nouns to which this token acts as modifier (<i>ncmod</i> relation)
the lemmas of the modifiers of this token (<i>ncmod</i> relation)

3 Results and analysis

For each of the experiments we used the CRF implementation of MALLET and trained the model until convergence. During testing, we followed the same preprocessing and feature extraction procedure, with the exception that we didn't use the boundaries of the gene mentions for tokenization since they were unknown. The results for the three submitted runs appear in Table 3. For our first run, we trained a 3rd order CRF model on the standard RASP tokenizer's output. For the second run, we altered the tokenization step in order to include dashes and slashes as token separators, since, according to the annotation scheme, in cases such as "*p65-selected*", only "*p65*" should be returned as a gene mention. This improved the performance substantially. For the third run, we kept the tokenization from the second run, but we reduced the CRF order to second order, since we would

like to reduce the training time of the system. There was a slight increase in performance, probably because the lower order CRF looks for simpler patterns which resulted in better recall.

Table 3: Evaluation of the submitted runs

	Precision	Recall	F
Run1	85.37	74.11	79.34
Run2	86.59	79.15	82.70
Run3	86.28	79.66	82.84

We also wanted to explore how beneficial was the use of linguistic features. Therefore, using Run3 as the basis (2nd order CRF with adapted tokenization), we ran experiments with subsets of the features extracted from the output of RASP. The results of Table 4 suggest that lemmas appear to be the most useful features, while POS tags and syntactic features improve performance less. One should take into account though that, apart from the noise introduced during parsing, specific syntactic features are only useful in sentences that exhibit them. For example, in the sentence “*For the P transcript from phage with the G(-) orientation...*”, “*P transcript*” is a gene mention but the lemmas “*transcript*” and “*p*” are not strong enough cues since they can be found outside of gene mentions. As a result, the model without syntactic features fails to recognize it as such. However, when the fact that “*p*” is a modifier of “*transcript*” is added as a feature from the syntactic analysis of RASP, then it is recognized correctly. In order to demonstrate the usefulness of the syntactic features more clearly, there needs to be an evaluation on an appropriate test set that contains more cases that need such features. Also, consistent annotation of the test set is important for quantitative assessment. In order to demonstrate this point, we measured our performance using only the first set of annotations (GENE.eval). As column F-strict of Table 4 shows, while the scores are lower, the gains in performance obtained by adding more features are larger than those observed when evaluating using both sets of annotations.

Table 4: Evaluation of the features

features	Precision	Recall	F	F-strict
simple_features	82.97	76.64	79.68	66.55
simple_features+lemmas	86.13	79.56	82.72	70.85
simple_features+lemmas+pos	85.82	79.91	82.76	71.03
simple_features+lemmas+pos+syntax	86.28	79.66	82.84	71.55

References

- [1] Briscoe, E., Carroll, J. and Watson, R., The Second Release of the RASP System, *In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, 2006.
- [2] Lafferty, J. D., McCallum, A. and Pereira, F. C. N., Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, *Proceedings of ICML 2001*, 282–289, 2001.
- [3] McCallum A. K., MALLET: A Machine Learning for Language Toolkit, <http://mallet.cs.umass.edu>
- [4] Settles B., Biomedical Named Entity Recognition Using Conditional Random Fields and Novel Feature Sets, *Proceedings of the JNLPBA*, 2004.



Named Entity Recognition with Combinations of Conditional Random Fields

Roman Klinger

roman.klinger@scai.fhg.de

Christoph M. Friedrich

christoph.friedrich@scai.fhg.de

Juliane Fluck

juliane.fluck@scai.fhg.de

Martin Hofmann-Apitius

martin.hofmann-apitius@scai.fhg.de

Fraunhofer Institute for Algorithms and Scientific Computing (SCAI)

Department of Bioinformatics

Schloss Birlinghoven, 53754 Sankt Augustin, Germany

Abstract

The *Gene Mention* task is a *Named Entity Recognition* (NER) task for labeling gene and gene product names in biomedical text. To deal with acceptable alternatives additionally to the gold standard, we use combinations of *Conditional Random Fields* (CRF) together with a normalizing tagger. This process is followed by a postprocessing step including an acronym disambiguation based on Latent Semantic Analysis (LSA). For robust model selection we apply 50-fold *Bootstrapping* to obtain an average F-Score of 84.58 % on the trainingset and 86.33 % on the test set.

Keywords: named entity recognition, text mining, data mining, conditional random fields, multi model approach

1 Introduction

In general, machine-learning solutions deal with a single truth. One characteristic in BioCreative 2006 compared to common NER tasks is that the training data contains acceptable alternatives for gene and protein names next to the gold standard. One problem using only the gold standard is that this information is possibly more ambiguous than necessary. For example in the sentence “*On the other hand factor IX activity is decreased in coumarin treatment with factor IX antigen remaining normal.*”

the gold standard is the twice annotation of *factor IX*. The alternative annotation gives the information that finding *factor IX antigen* is just as well. But in “*The arginyl peptide bonds that are cleaved in the conversion of human factor IX to factor IXa by factor XIa were identified as Arg145-Ala146 and Arg180-Val181.*” the gold standard is finding *human factor IX* and *factor IXa* and *factor XIa* but the alternative gives us the possibility of *factor IX* instead of *human factor IX*.

We address that problem with a multi model approach using the Conditional Random Fields [5] implementation *Mallet* [7] which showed superior results in BioCreative 2004 [4] and our previous works [2].

2 System Description

The developed system is inspired by [8, 9]. A sketch of the workflow can be found in figure 1. At first the external tools *GeniaTagger* [11] and *ProMiner* [3] are called. Their results are used as IOB-features, which form the input for *Mallet* to build multiple Conditional Random Fields together with the sentences (in the file *train.in*) and the annotation information (in files *GENE.eval* and *ALTPGENE.eval*).

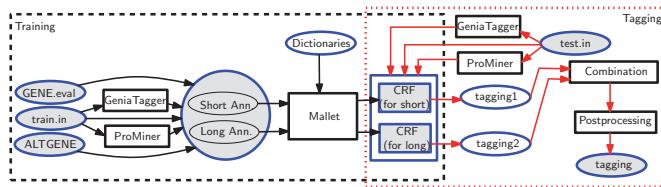


Figure 1: Workflow of our system.

Table 1: Strategies to combine different annotations. For the examples let us assume to have *fibrinogen degradation products* as annotation from the model trained on long annotations and *fibrinogen* and *FDP* as annotations from the model trained on short annotations on the text part *fibrinogen degradation products (FDP)*.

No.	Strategy	Example
1	Use long annotation first, then add short annotation (without overlaps)	<i>fibrinogen degradation products ; FDP</i>
2	Use short annotation first, then add long annotation (without overlaps)	<i>fibrinogen ; FDP</i>
3	Greedy: Combine both (with overlaps)	<i>fibrinogen ; FDP ; fibrinogen degradation products</i>

These multiple models deal with mentioned ambiguities by building one annotation out of the shortest possibilities and one out of the longest ones, each without overlaps. In the first example sentence mentioned in the introduction the short annotations are the ones from the gold standard, but in the second sentence we would use *factor IX* instead of *human factor IX*.

The generated models can then be used for tagging the new sentences (here we assume them to be in the file test.in) followed by the combination of these different outputs. We tried three different methods displayed in table 1. In the example in the second method nothing is added because the long annotation overlaps the short annotation.

The last step is postprocessing. Unequal numbers of closing or opening brackets are corrected and an acronym disambiguation using latent semantic analysis is conducted. It concerns the high frequent ambiguous acronyms *CAD*, *CSF*, *REM* or *CAP*. This concept study works here only at the sentence level but can be shown to be more powerful, if the full sentence context will be available.

3 Analysis and Results

For selecting the training parameters of the conditional random fields we use bootstrapping [1] with 50 replicates having approximately 9480 training and 5520 validation examples in every replicate.

In our rich feature set we have different types of features like morphological [8] (some automatically generated [10]), dictionaries ([6] and self-made), offset conjunction and part-of-speech/shallow parsing information from the *GeniaTagger* [11]. Additionally we use the tagging information of the *ProMiner* [3], which achieves a precision of 0.88 on the training and 0.87 on the test set but a lower recall.

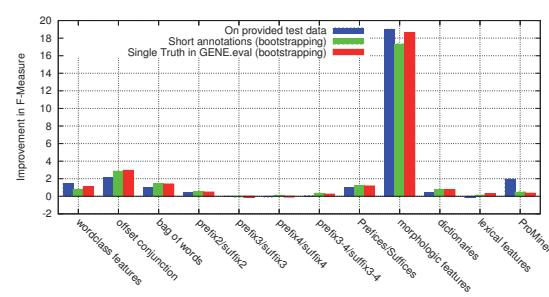


Figure 2: Influence of features estimated by omitting them.

detect an higher importance of using prefixes and suffixes of all lengths (2–4) in comparison to only using these with length 2. This is not expectable because prefixes and suffixes of length 3 and 4 have no impact. This is an example for features not being independent as can also be seen in figure 2. It is not possible to have a greedy analysis of all combinations of the features because of prohibitive training times (about 1–2 hours, depending on the size of the feature set). Instead we conducted a systematic feature analysis based on attribute groups.

Analysing the tokenisation, we started with a complex tokenisation (inspired by [9]) reaching a mean F-score of 0.821 (using bootstrapping) on the system trained on the gold standard information in GENE.eval.

Two classes of parameters are most important: The combination and selection of features and the tokenisation of the text. The impact of each feature or group of features was computed by building a conditional random field without them. It is displayed in figure 2. Morphological features are of overwhelming importance followed by offset conjunction. Interestingly the ProMiner has only a minor impact on the training set but improves results on the test set (with 2%). So it can be concluded that the performed approximative search has a higher impact than the simple dictionary matching.

The improvement of combinations of features is complex as can be seen in the case of prefixes. We

Table 2: Results on the trainingset (averaged over 50 bootstrap replicates) and on the test set after postprocessing and disambiguation (Standard deviation is given in brackets, submitted runs are marked with a *).

Model	Bootstrapping on Trainingset			On Testset		
	Precision	Recall	F-Score	Precision	Recall	F-Score
Long	86.30 (0.0065)	79.53 (0.0094)	82.78 (0.0064)	87.41	80.29	83.70
Short*	86.87 (0.0054)	81.94 (0.0106)	84.33 (0.0069)	88.57	83.83	86.13
Greedy*	80.21 (0.0069)	89.47 (0.0057)	84.58 (0.0047)	82.02	90.63	86.11
Long first*	85.38 (0.0060)	83.63 (0.0079)	84.50 (0.0055)	87.27	85.41	86.33
Short first	83.83 (0.0063)	84.81 (0.0065)	84.32 (0.0048)	85.50	85.61	85.56
GENE.eval	86.61 (0.0071)	81.76 (0.0123)	84.11 (0.0076)	87.86	83.53	85.64

Splitting always on dashes improves the results to 0.835.

The results of our systems on the training set and on the test set are displayed in table 2. The ratios between the experiments on the test data and on the training data with different models are similar, so we can assume bootstrapping as an appropriate choice for model selection. We see that it is already useful only to select a special subset of the alternatives for training: The annotation made by the short expert yields in better results than the one in GENE.eval. The combination of the short and long ones has further impact dependent on the strategie: Adding the short and long annotation the greedy way yields in a very high recall but a lower precision than the other methods. Using the long annotation and adding the short one gives us an higher precision and the highest F-Score of the different strategies.

References

- [1] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.
- [2] Ch. M. Friedrich, T. Revillion, M. Hofmann, and J. Fluck. Biomedical and chemical named entity recognition with conditional random fields: The advantage of dictionary features. In S. Ananiadou and J. Fluck, editors, *Proceedings of the Second International Symposium on Semantic Mining in Biomedicine (SMBM 2006)*, pages 85–89, 2006.
- [3] D. Hanisch, K. Fundel, H. T Mevissen., R. Zimmer, and J. Fluck. ProMiner: Organism-specific protein name detection using approximate string matching. In *Proceedings of the BioCreative Challenge Evaluation Workshop 2004*, 2004.
- [4] L. Hirschman, A. Yeh, Ch. Blaschke, and A. Valencia. Overview of biocreative: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6 Suppl 1:S1, 2005.
- [5] J.D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 282–289. Morgan Kaufmann Publishers, 2001.
- [6] K. Lerman, Y. Jin, E. Pancoast, and R. McDonald. Biotagger. Software.
- [7] A. K. McCallum. MALLET: A Machine Learning for Language Toolkit. <http://mallet.cs.umass.edu>, 2002.
- [8] R. McDonald and F. Pereira. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, 6 (Suppl 1)(S6), 2005.
- [9] B. Settles. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192, 2005.
- [10] B. Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*, Geneva, Switzerland, 2004.
- [11] Y. Tsuruoka, Y. Tateishi, J.-D. Kim, T. Ohta, J. McNaught, Sophia Ananiadou, and Jun’ichi Tsujii. Developing a robust part-of-speech tagger for biomedical text. In *Advances in Informatics - 10th Panhellenic Conference on Informatics, LNCS 3746*, pages 382–392, 2005.

Acknowledgements

This work has been partially funded by the MPG-FhG Machine Learning Collaboration
<http://lip.fml.tuebingen.mpg.de/>



Gene Mention Recognition Using Lexicon Match Based Two-Layer Support Vector Machines

Yifei Chen¹

yifechen@vub.ac.be

Feng Liu¹

fengliu@vub.ac.be

Bernard Manderick¹

bmanderic@vub.ac.be

¹ Computational Modeling Lab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

1 Method

1.1 System Description

In the Gene Mention Tagging task of BioCreAtIvE II, two Support Vector Machines (SVM) and a post-processor are proposed to compose our two-layer gene mention recognition system (see Figure 1). The first layer is a text to gene mention layer, which takes original texts as inputs and predicts gene mentions. The second layer is a gene mention to gene mention layer, which takes predicted gene mention tags from the first layer as inputs and outputs the final tags.

In addition, we also incorporate an ensemble of post-processing modules as a post-processor: an abbreviation resolution module, a boundary check module and a name refinement module, into the system to further improve the performance.

1.2 Data Representation

In this task, 15,000 sentences were prepared, from which we take 80% sentences as training data and the rest as development test data to implement a 5-fold cross validation for system development.

The tokenization strategy in our system is to split the sentences based on spaces and punctuations. Then we use the traditional BIO representation to tag each token.

- B: current token is the beginning of a gene mention
- I: current token is inside a gene mention
- O: current token is outside any gene mention

For example, a sentence “Takayasu’s disease: association with HLA-B5.” with “HLA-B5” as a gene mention can be represented as “Takayasu/O /O s/O disease/O :/O association/O with/O HLA/B -/I B5/I ./O”.

1.3 Feature Extraction

In the first layer, the following features are extracted for each token.

- Token: Current token itself.
- Lexicon match: Matching the current token and its surrounding tokens against gene mention lexicon entries. We employ a closed lexicon in the system, which is constructed from the training data by collecting all the terms that are annotated as gene mentions. Uni-, bi-, tri-grams of tokens surrounding the current token are provided to match the lexicon entries using strict and partial matching strategies respectively.

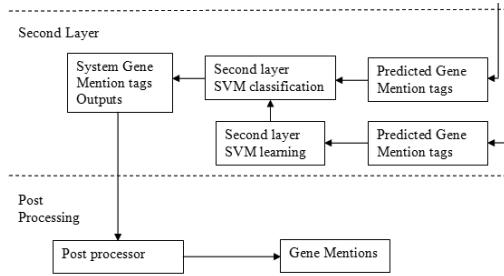


Figure 1: System Description

- Orthographic features: Table 1 shows all the orthographic features.

Table 1: Orthographic features.

Feature	Example	Feature	Example	Feature	Example
DigitNumber	10	LeftParenthesis	(Percent	%
GreekAlphabet	alpha	RightParenthesis)	Comma	,
SingleCapital	T	Hyphen	-	Period	.
InitialCapital	Rho	Backslash	/	Article	the
LowerMixCapital	GnRH	LeftBracket	[Conjunction	and
AlphabetMixDigit	p26	RightBracket]	RomanNumber	IV
AllCapitals	SGPT	Colon	:	Others	?
AllLowers	stonin	SemiColon	;		

- Prefix and Suffix: Bi-, tri- and quad- prefix and suffix of the current token.
- POS: Part of speech of the current token. We choose the MedPost tagger [3] to tag both the training and test data in our system.

In the second layer, only the class label feature is used, which is the predicted tag of the current token from the first layer. So this feature has three values: B, I and O.

Both layers employ a sliding window strategy to introduce neighboring knowledge of the current token. According to the different effects that surrounding tokens give to the current token, window sizes can be selected respectively for the different layers.

1.4 Support Vector Machine

Support Vector Machine (SVM) [4] is proposed to train the above two-layer model. SVM is a powerful machine learning method, which has been applied successfully in the named entity recognition

domain [2]. In this competition, we choose the toolbox LIBSVM [1], a java/C++ library for SVM learning. We adopt a polynomial kernel with $\text{degree} = 2$ and $\text{coefficient} = 0$.

1.5 Post-processing

In order to improve the performance further, we develop an ensemble of post-processing modules. The abbreviation resolution module can recover the errors caused by incorrectly mapping abbreviations to their full forms. The boundary check module can recover the boundary errors caused by our tokenization strategy, BIO representation and missing trigger words. The name refinement module employs some rules to refine the recognized gene mentions by removing the redundancy and inconsistency.

2 Results and Analysis

Finally, our two-layer gene mention recognition system is trained on the whole training data (15,000 sentences), and tested on the novel test data (5,000 sentences). The performance of the system in this competition is shown in Table 2.

Table 2: Results of the two-layer gene mention recognition system.

	Precision	Recall	$F_{\beta=1}$
the first layer	76.90	66.71	71.50
the second layer	81.50	68.45	74.40
after post-processing	88.83	69.70	78.11

From Table 2 we can see that from the first layer to the second layer, the precision, recall and $F_{\beta=1}$ score are increased by 4.6%, 1.74% and 3.1% respectively, which means that the second layer can improve the performance by introducing the proceeding and succeeding tags information of the current token. Post-processing can also increase these three values, especially increasing the precision by 7.33%. Therefore our post-processor can recover the false positive errors effectively.

After the competition, our team are still working on developing the appropriate strategy to build and match gene mention lexicon entries, and dealing with the low recall caused by gene mention spelling variation to achieve the better $F_{\beta=1}$ score.

Acknowledgement

The first author is funded by a doctoral grant of the Fonds voor Wetenschappelijk Onderzoek (FWO) and the second author is funded by a doctoral grant of the Vrije Universiteit Brussel (VUB).

References

- [1] Chang, C.-C. and Lin, C.-J. *LIBSVM*: a library for support vector machines, 2001.
 - [2] Kazama J, Makino T, Ohta Y, Tsujii J, Tuning Support Vector Machines for Biomedical Named Entity Recognition, *Proceedings of the workshop on Natural Language Processing in the Biomedical Domain (at ACL'2002)*, 1–8, 2002.
 - [3] Smith, L., Rindflesch, T. and Wilbur, W.J., Medpost: a part-of-speech tagger for biomedical text, *BIOINFORMATICS*, 20 (14): 2320–2321, 2004.
 - [4] Vapnik, V., *The nature of statistical learning theory*, Springer-Verlog, New York, 1995.



Using Semi-Supervised Techniques to Detect Gene Mentions

Sophia Katrenko¹ Pieter W. Adriaans¹

katrenko@science.uva.nl pietera@science.uva.nl

¹ Human Computer Studies Laboratory, Institute of Informatics,
University of Amsterdam, Kruislaan 419, 1098VA Amsterdam, the Netherlands

Abstract

This report presents an approach which has been taken to detect gene mentions (Gene Mention Tagging task of BioCreAtIvE II). We investigate the semi-supervised learning techniques, in particular, co-training (with orthographic/contextual features split) and self-training using a subset of GENIA corpus as a pool of unlabeled data.

Keywords: co-training, self-training

1 Introduction

Learning with unlabeled data constitutes an attractive alternative to the supervised techniques. First, it is possible to reduce the size of the initial labeled training set and, consequently to avoid time-consuming annotating process. Second, it is useful to study which limits the semi-supervised learning has and if adding more unlabeled data to the initial training set can result in performance comparable to the supervised techniques. Semi-supervised methods have been already successfully applied to the number of tasks, including image recognition, word sense disambiguation, named entity recognition.

2 Data Preprocessing and Methods

Given BioCreAtIvE II training and test data sets, both of them were tokenized. No further analysis of the data (PoS tagging, parsing) has been performed.

While constructing feature set, we focused on the features which can be easily extracted from the text data and features already used to perform the named entity recognition task in the past. We have not considered any pre-compiled lists of proteins, even though such resources could have improved the performance greatly. In spite of widely recognized usefulness of other information such as stop word lists or post-processing steps [2], it was not included into our system either. Our intention was therefore to keep the feature set as small as possible (yet sufficient for the gene mentions recognition task) and to focus on the impact of the unlabeled data instead. We used both contextual features (constructed by the window of two tokens to the left and to the right of the token in focus) and orthographic (such as presence of digits, hyphens in a token, capitalization and some others).

As an underlying classifier we used conditional random fields method (CRFs) [1] which is a probabilistic framework employing conditional probability over the sequences of labels. It is particularly suited for labeling sequential data and already proved to be successful when detecting named entities in the biomedical domain [1]. Further, we selected co-training and self-training as two techniques widely used in the semi-supervised setting. Co-training [4] is a machine learning method which uses an idea of two distinct views on the data examples. Two learning algorithms are then used, where

each of them is trained on one view only but also receives the most confident examples labeled by the other. In each iteration, the training set is enlarged by the predictions made by two classifiers.

Self-training is the simplest form of the semi-supervised learning, where a single classifier labels the unseen examples in each iteration and adds the most confident to its training set. We submitted two runs, one based on the self-training results (*run1*) and the other whose results were obtained by co-training (*run 2*).

3 Analysis

Self-training In the self-training setting we carried out several experiments by varying the number of iterations, the size of the training set and the size of the unlabeled pool. The run we submitted had the following settings: number of iterations is equal to 5, number of instances added in each iteration is 100, and 1,000 Medline sentences from GENIA corpus [5] are used as a source of the unlabeled data. Labeled examples have always been sampled from the training data set provided by the organizers of BioCreAtIvE II. In each iteration, only the most confident predictions are added (top N). In this setting precision is much higher than recall (82,28% versus 71,08%) and F-score equals 76.27%.

To compare how self-training behaves in each iteration, we applied the models created in each of these iterations to the test set. We did not observe any significant changes in performance from one iteration to another. One possible explanation for this would be either adding the instances from the unlabeled data which are already in the training set, or adding new protein names and some amount of noise. To verify the latter, we checked all of the labeled instances added from the unlabeled pool. Such check became possible because we use the GENIA corpus, which has already been annotated with several biomedical types, such as *protein*, *cell type*, *cell line*, *RNA*, and *DNA*.

It is reasonable to assume that the corpus used in the Biocreative II challenge has been annotated according to some other guidelines than the GENIA corpus. Nevertheless, the GENIA annotation can give us more insight on what happens by adding newly labeled instances in each iteration. In particular, it is possible to check how many unique predictions are added and how many of them are true positives. For instance, after 5 iterations of self-training using 60% of the training set, 241 predictions were added to the initial training set from the subset of GENIA (1,000 sentences). 81,33% of these predictions are annotated in GENIA as protein names. Boundaries of only 12,24% of the true positives added to the training set do not exactly correspond to the annotation used in GENIA. The detected protein names are of different length, up to the named entities consisting of 7 tokens. We can conclude that although the overall precision of unlabeled examples added to the training set is relatively high, there is also noise added.

Interestingly, reduction of the labeled data does not significantly affect precision (Table 1) (in all experiments it is around 80%). In contrast, recall can be boosted either by adding more labeled examples or by using much larger pool of unlabeled instances (Table 2). The semi-supervised methods meant to be applied when the size of labeled data set is much smaller than the size of the unlabeled data set.

Co-training Although our initial expectation was to receive better performance using co-training than self-learning, it was not supported by the experiments in practice. As discussed in [3], performance in the former case crucially depends on the correlation between the feature sets used to train classifiers. We split the feature set in two subsets, contextual and orthographic features.

In the co-training setting, the number of iterations was set to 6. Surprisingly, self-training outperformed co-training (F-score dropped to 71,74%). Co-training nevertheless provides better results than applying contextual and orthographic models separately. To verify that the initial classifiers are independent, we used the Φ^2 statistics as described in [3]:

$$\Phi^2 = \sum_{i=0}^1 \sum_{j=0}^1 \frac{(P(E_1 = i, E_2 = j) - P(E_1 = i) * P(E_2 = j))^2}{P(E_1 = i) * P(E_2 = j)}$$

Here, E_1 and E_2 are two random variables, whose values depend on whether a classifier errs on an unlabeled example (0) or does not (1). If E_1 and E_2 are independent, $\Phi^2 = 0$. It is argued in [3] that, given the data used in [3], co-training is always beneficial for $\Phi^2 < 10\%$ and harms performance if $\Phi^2 > 60\%$. In our case, $\Phi^2 = 21\%$, which might explain why using co-training results in only slightly higher performance.

Table 1: Effect of the size of the training set on performance (on the test set)

Size of the training set	Precision	Recall	F-score
3,750 sentences	80,5%	45,92%	58,48%
9,000 sentences	83,64%	49,66%	62,32%
15,000 sentences	81,17%	72,45%	76,56%

Table 2: Effect of the size of the unlabeled pool on performance (on the test set)

Size of the unlabeled data set	Precision	Recall	F-score
1,000 abstracts	83,64%	49,66%	62,32%
2,000 abstracts	80,25%	64,87%	71,72%

4 Future Work

For the future research it will be particularly interesting to explore how the predictions of a given confidence affect performance. In our case, only the top N predictions were added to the training set and it might be the case that predictions of a bit lower confidence would provide more useful information.

We also plan to compare results received by co-training and self-training in more detail. In particular, we would like to address the problem of boosting precision. As it can be seen from the results presented above, recall can be boosted by adding more labeled and/or unlabeled examples. Since precision does not change very much, our assumption is that re-considering the current feature set might help to get better results in terms of precision.

References

- [1] Settles, B. ABNER: an open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics*, 21(14):3191-3192, 2005.
- [2] Hakenberg, J., Bickel, S., Plake, C., Brefeld, U., Zahn, H., Faulstich, L., Leser, U., and Scheffer, T., Systematic feature evaluation for gene name recognition. *BMC Bioinformatics* 6(1), 2005.
- [3] Krogel, M.-A., and Scheffer, T., Multirelational learning, text mining, and semi-supervised learning for functional genomics. *Machine Learning* 57(1/2):61-81, 2004.
- [4] Blum, A., and Mitchell, T., Combining Labeled and Unlabeled Data with Co-Training, *Proc. of the 1998 Conference on Computational Learning Theory*, July 1998.
- [5] Jin-Dong, K., Ohta, T., Teteisi, Y., and Tsujii, J. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl. 1), pp. 180-182, Oxford University Press.



BioCreative II Gene Mention Tagging System at IBM Watson

Rie Kubota Ando

rie1@us.ibm.com

IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532, USA

Abstract

This paper describes our system developed for the BioCreative II gene mention tagging task. The goal of this task is to annotate mentions of genes or gene products in the given Medline sentences. Our focus was to experiment with a semi-supervised learning method, *Alternating Structure Optimization* (ASO) [1], by which we exploited a large amount of *unlabeled data* in addition to the labeled training data provided by the organizer. The system is also equipped with automatic induction of high-order features, gene name lexicon lookup, classifier combination, and simple post-processing. Our system appears to be competitive. All of our three official runs belong to the Quartile 1.

1 Gene mention tagging system

Our gene mention tagging system was built on top of a named entity chunking system described in [1], which was used for annotating names of persons, organizations, and so forth. This system casts the chunking task into that of sequential labeling, as is commonly done, by encoding chunk information into token tags. It uses a regularized linear classifier with modified Huber loss and the 2-norm regularization. That is, using the ‘one-versus-all’ scheme, we train binary classifiers, one for each token tag, using n labeled data points $\{(\mathbf{x}_i, y_i)\}$ for $i = 1, \dots, n$ by: $\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} \sum_{i=1}^n L(\mathbf{w}^T \mathbf{x}_i, y_i) + \lambda \|\mathbf{w}\|^2$. The regularization parameter λ is set to 10^{-4} . L is the loss function: $L(p, y) = \max(0, 1 - py)^2$ if $py \geq -1$; and $-4py$ otherwise. The optimization is done by stochastic gradient descent. Viterbi-style dynamic programming is performed to find the token tag sequence with the largest confidence. Feature types are shown in Figure 1. Using this framework, we experimented with additional resources and algorithms, which we describe below.

-
- words, parts-of-speech, character types, 4 characters at beginning/ending in a 5-word window
 - words in a 3-syntactic chunk window.
 - labels assigned to two words on the left.
 - bi-grams of the current word and the left label.
 - labels assigned to previous occurrences of the current word.
-

Figure 1: Feature types.

1.1 Exploiting unlabeled data through Alternating Structure Optimization (ASO)

ASO is a multi-task learning algorithm that seeks to improve performance on individual tasks by simultaneously learning multiple tasks that are related to each other. The application of ASO to semi-supervised learning involves automatic generation of thousands of prediction problems (called ‘auxiliary problems’) and their labeled data from unlabeled data, so that the multi-task learning algorithm can be applied on the unlabeled data.

To put this into perspective, ASO-based semi-supervised learning can be viewed as learning new (and better) feature representation from unlabeled data. This is done by learning auxiliary predictors that predict one part of the feature vectors from another part of the feature vectors, which can be learned from unlabeled data. Under certain conditions, it can be shown that learning auxiliary predictors of this type can reveal the predictive structure (something useful for the target prediction problems) underlying the data. The final classifiers are trained with labeled data using the original features and the new features learned from unlabeled data. Since modern classifiers based on empirical risk minimization are capable of ignoring irrelevant features to some degree, the risk of using unlabeled data this way is relatively low, and its potential gain is large. [1] should be consulted for the details of ASO. Below, we only describe the specifics of our setting.

Auxiliary problems The ‘word prediction’ auxiliary problems were used with the same implementation details as in [1].

Unlabeled data For unlabeled data, we had about 5 million Medline abstracts (consisting of approx. 500 million words) over a 10-year period (1994–2003) at hand. To utilize the entire corpus through ASO, the training of thousands of predictors on these 500 million words were required, which we felt would be too resource-intensive. However, our experiments using the old BioCreative I data set indicated that if we use a small subset of the unlabeled corpus generated by random sentence selection, the performance improvement from ASO was marginal. This was due to the small size of the vocabulary overlap between the unlabeled data and the training/test data. (This issue appears to be specific to the biomedical domain, which has a much larger vocabulary than, for instance, the news domain.) To benefit from unlabeled data with reasonable computation time, we created a small but useful unlabeled corpus as follows. We go through every sentence of the input corpus while counting up the occurrences of words. If the sentence contains at least one word that has occurred less than k times so far, then we choose this sentence; discard it otherwise. By setting $k = 25$, we obtained an unlabeled corpus that is much smaller than the original one but represents well (to some degree) the entire vocabulary of the original corpus.

1.2 Automatic induction of high-order features

High-order features (combining two or more base features) are sometimes effective, but generating all the combinations would make training expensive. We used a simple method for selectively generating bi-gram features. The idea is to select a bi-gram feature only if it would help to correctly classify the data point that was misclassified when only base features (in Figure 1) were used. This is done as follows. First, we train a classifier using the base features only on a labeled data set L_1 . Next, we test this classifier on a labeled data set L_2 . We generate bi-gram features (e.g., ‘current-word=“gene” and next-word=“(”) only from the data points that are misclassified. We filter out all but those occurring in at least q misclassified data points. To further filter out the bi-grams, we consider $2K$ criteria for K -way classification, each of which inspects whether that bi-gram is useful as evidence for being positive/negative with respect to each class. According to each criteria, each bi-gram receives a score computed as a sum of partial derivatives of the loss function on the respective data points. The bi-gram is selected if its score is within top t according to one of the $2K$ criteria. We set $q = 10$ and $t = 100000$. Although one could divide the training set into L_1 and L_2 disjointly, we instead used the entire training set as L_1 and L_2 and applied an ‘early stop’ when training the classifier with base features.

1.3 Domain lexicons

Our two (out of three) official runs used a domain lexicon, which we generated from LocusLink, Swiss-Prot, and Mesh. Our domain lexicon consists of a list of names (e.g., “adenosine arabinose”) with tags that indicate the information source (e.g., “MESH”). In the feature generation process, we turn on the corresponding feature according to the tags associated with the matched name entries (including partial matching).

1.4 Classifier combination

A number of studies have shown that combining results of several classifiers (that, ideally, produce similar performance but make different mistakes) often improves performance over a single classifier. The classifiers to be combined could be, for instance, those employing different schemes of chunk encoding (e.g., one classifier uses BIO, and another uses EIO) or those based on different models (e.g., one is MaxEnt, and the other is SVM). [2] reported that combining a left-to-right chunker and a right-to-left chunker (by taking a union of the two sets of annotations) was effective on the BioCreative I data. We adopt this strategy and combine the results of a left-to-right chunker and a right-to-left chunker. However, instead of taking a union, we remove any annotation that overlaps with another by keeping longer ones, which performed better than taking a union in our experiments on the BioCreative I data.

	Post-processing	Feature induction	Name lexicons	Classifier combination	Unlabeled data	P	R	F
Baseline	–	–	–	–	–	89.13	79.39	83.98
Post-processing	X	–	–	–	–	89.40	79.39	84.10 (+0.12)
Feature induction	–	X	–	–	–	89.11	79.86	84.23 (+0.25)
Name lexicon	–	–	X	–	–	88.89	80.48	84.47 (+0.49)
Classifier combination	–	–	–	X	–	85.14	84.90	85.02 (+1.04)
Unlabeled data	–	–	–	–	X	91.17	81.52	86.07 (+2.09)
Run#3	X	X	X	–	X	91.54	81.99	86.50 (+2.52)
Run#1	X	X	–	X	X	88.37	85.94	87.14 (+3.16)
Run#2	X	X	X	X	X	88.48	85.97	87.21 (+3.23)

Figure 2: Performance results. Effectiveness of the five components; the three official runs. The best performance in each column is highlighted. The numbers in parentheses are performance improvements over the baseline (a supervised configuration using base features).

1.5 Simple post-processing

Many of the BioCreative I systems were equipped with some post-processing. We adopt the one used in [2], which removes annotations that include any unmatched parenthesis.

2 Performance results

Figure 2 shows the performance of our official runs and some post-submission experimental results. ‘Baseline’ is a standard supervised configuration using the features in Figure 1. The five rows following ‘Baseline’ compare the performance improvements obtained by the five components described above. The performance trend is consistent with that of our experiments on the BioCreative I data (not included in this paper). All the five components improved performance on the BioCreative II evaluation data as well as the BioCreative I data. The largest performance gain is obtained from unlabeled data through ASO. This semi-supervised configuration (‘Unlabeled data’) achieves 2.09 higher F-measure than the baseline. Also note that it outperforms the baseline both in precision and recall. The second best contributor is classifier combination, which improved recall at the price of precision and resulted in 1.04 improvement in F-measure.

Among the three official runs, Run#2 that uses all the five components achieved the best performance (3.23 higher than our baseline) among all of our configurations. These results clearly confirm the effectiveness of our approach on this task.

3 Conclusion

This paper presented the gene mention tagging system that participated in BioCreative II. The main strength of the system derives from semi-supervised learning using the ASO algorithm. We also experimented with classifier combination, domain lexicon, automatic generation of high-order features, and simple post-processing, which were all effective.

Since our approach is general, we expect it to be also useful for tagging other types of mentions in the biomedical text. We presume that semi-supervised learning is particularly suitable for exploring biomedical texts, given the presence of a huge amount of unlabeled data – the Medline corpus.

References

- [1] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.
- [2] S. Dingare, J. Finkel, C. D. Manning, M. Nissim, and B. Alex. Exploring the boundaries: Gene and protein identification in biomedical text. In *Proceedings of the BioCreative Workshop*, 2004.



Rich Feature Set, Unification of Bidirectional Parsing and Dictionary Filtering for High F-Score Gene Mention Tagging

Cheng-Ju Kuo¹

cju.kuo@gmail.com

Yu-Ming Chang²

porter@iis.sinica.edu.tw

Han-Shen Huang²

hanshen@iis.sinica.edu.tw

Kuan-Ting Lin²

woody@iis.sinica.edu.tw

Bo-Hou Yang^{2,3}

ericyang@iis.sinica.edu.tw

Yu-Shi Lin²

bathroom@iis.sinica.edu.tw

Chun-Nan Hsu²

chunnan@iis.sinica.edu.tw

I-Fang Chung¹

ifchung@ym.edu.tw

¹ Institute of Bioinformatics, National Yang-Ming University, Taipei, Taiwan

² Institute of Information Science, Academia Sinica, Taipei, Taiwan

³ Department of Electrical Engineering, Chang-Gung University, TaoYuan, Taiwan

In the first BioCreative (2004) [3], conditional random fields (CRFs) [5] were employed in tagging gene and protein mentioned in the biomedical text with high performance [8]. Therefore, we chose CRFs as our starting point and carefully selected a rich set of 5,059,368 predicates as the features. To further improve its performance, we combined the tagging results of forward and backward parsing [4]. We tried different combination methods, including set operations and Co-Training [1]. However, we found that Co-Training performed poorly. Instead, we selected the best solutions from the “adjacent” ten candidates of bidirectional parsing and then applied dictionary filtering to obtain the best F-score result. Details are given as follows.

We applied MALLETT [7] to take advantage of its feature induction capability [6]. Due to the special characteristics of name-entities of genes and gene products [10], a rich set of features is required. Not all features proposed in previous work are useful. After hundreds of trials, we carefully selected predicates shown in Table 1 as our feature set, which includes commonly used orthographic predicates and character-n-gram predicates for $2 \leq n \leq 4$ [8]. We used $\{-2, -1, 0, 1, 2\}$ as the offsets and evaluated predicates such as word, stemmed word, part-of-speech tag, and word morphology as the contextual features at each position. Our domain-specific features include nucleotide (i.e., types of DNA or RNA), residues of amino acids, etc. We excluded prefix and suffix predicates used in previous work because we found that they usually increase false positive. To extract features, the Genia Tagger [9] was applied for stemming, tokenization and part-of-speech tagging. We modified the Genia Tagger slightly to tokenize words with a higher granularity. For example, punctuation symbols within words were segmented. We also applied a rule-based filter to clean up some easily fixed mistakes, such as entities with unpaired parentheses or square brackets.

The performance of the CRF models with this feature set and the rule-based filter is given in the first row of Table 2, which is already slightly better than previously reported figures. These inside test results were obtained by randomly selected 10,000 sentences for training and the rest for testing from the training data set provided by the organizers. To further improve its performance, we combined the tagging results of forward and backward parsing. In forward parsing, the tagger reads and tags the input sentences from left to right, while in backward parsing, the tagger reads and tags the input sentences from right to left. Note that the training set and the features must be reversed to train a backward parsing CRF model. We tested the forward and backward parsing models and found that backward parsing constantly outperformed forward parsing in both recall and precision, but its

Table 1: Features.

Feature	Example	Feature	Example	Feature	Example
Word	proteins	Hyphen	-	Nucleoside	Thymine
StemmedWord	protein	BackSlash	/	Nucleotide	ATP
PartOfSpeech	NN	OpenSquare	[Roman	I, II, XI
InitCap	Kinase	CloseSquare]	MorphologyTypeI	p53→p*
EndCap	kappaB	Colon	:	MorphologyTypeII	p53→a1
AllCaps	SOX	SemiColon	;	MorphologyTypeIII	GnRH→AaAAA
LowerCase	interlukin	Percent	%	WordLength	1, 2, 3-5, 6+
MixCase	RalGDS	OpenParen	(N-grams(2-4)	p53→{p5, 53}
SingleCap	kDa	CloseParen)	ATCGUsequence	ATCGU
TwoCap	IL	Comma	,	Greek	alpha
ThreeCap	CSF	FullStop	.	NucleicAcid	cDNA
MoreCap	RESULT	Apostrophe	'	AminoAcidLong	tyrosine
SingleDigit	1	QuotationMark	"	AminoAcidShort	Ser
TwoDigit	22	Star	*	AminoAcid+Position	Ser150
FourDigit	1983	Equal	=		
MoreDigit	513256	Plus	+		

Table 2: System performance in inside test.

System	Precision	Recall	F-Measure
Forward	0.8660	0.8077	0.8359
Backward	0.8733	0.8118	0.8414
Union	0.8349	0.8578	0.8462
Intersection	0.9076	0.7186	0.8021
Adjacent Ten Union + Dictionary	0.8773	0.8263	0.8510

reason is unclear. We assume that some “signals” at the end of entities are more important to well demarcate boundaries of entities. However, distributions of nonzero features in both parsing directions show no significant difference (data not reported here). Then, we tried different ways to combine the bidirectional tagging results. Simple set operations failed to improve the performance. Though recall may be enhanced by union and precision by intersection, they also degraded the other measure and the F-score. Table 2 shows their inside test F-scores. We tried to apply Co-Training [1]. However, since the output scores (negative log likelihood) of Mallet were not reliable to select unlabeled training data, Co-Training seriously degraded the F-score to as low as 0.6.

Meanwhile, we found that the union of the “adjacent” ten tagging solutions of bidirectional parsing may achieve a nearly perfect recall (0.9810 for the final test, with 0.1387 precision). That is, nearly all true positives are in this union. The “adjacent” solutions were obtained by Mallet’s *n-best* option. However, we found that the solutions are not actually the best *n* solutions. Instead, they are candidate tagging results adjacent to the best tagging in the search tree grown by the *A** search algorithm, according to our trace of Mallet’s source code. This also explains why its output score ranking is not appropriate for Co-Training. In fact, exhaustively search for the best *n* candidates is intractable. Nevertheless, knowing that nearly all true positives are actually in the union of the adjacent ten solutions, we distill real true positives from this union as follows.

1. Parse the input sentence in both directions to obtain the adjacent ten solutions for each direction

Table 3: System performance of submitted runs.

System	Precision	Recall	F-Measure
Backward	0.8930	0.8383	0.8648
Union	0.8610	0.8708	0.8658
Adjacent Ten + Dictionary	0.8930	0.8449	0.8683

with their output scores;

2. Compute the intersection of bidirectional parsing and select the solution in the intersection that minimizes the sum of its output scores;
3. For the other 18 solutions, select the labeled terms appearing in a dictionary with its length greater than three.

We used approved gene symbols and aliases obtained from HUGO [2] as our dictionary for the final dictionary filtering. We submitted the results of the top three performing methods in our inside test (see Table 2) for the 2nd BioCreative (2006). Their performances are shown in Table 3.

References

- [1] Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, Morgan Kaufmann Publishers, pages 92–100, 1998.
- [2] Eyre, T. A., Ducluzeau, F., Sneddon, T. P., Povey, S., Bruford, E. A., and Lush, M. J. The HUGO gene nomenclature database, 2006 updates. *Nucleic Acids Research*, 34:D319–D321, 2006.
- [3] Hirschman, L., Yeh, A., Blaschke, C., and Valencia, A. Overview of BioCreAtIVe: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6:S1, 2005.
- [4] Kudo, T. and Matsumoto, Y. Chunking with support vector machines. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001.
- [5] Lafferty, J., McCallum, A., and Pereira, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
- [6] McCallum, A. Efficiently inducing features of conditional random fields. In *Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03)*, 2003.
- [7] McCallum, A. K. Mallet: A Machine Learning for Language Toolkit. <http://mallet.cs.umass.edu>, 2002.
- [8] McDonald, R. and Pereira, F. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, 6:S6, 2005.
- [9] Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., and Tsujii, J. Developing a Robust Part-of-Speech Tagger for Biomedical Text. In *Advances in Informatics - 10th Panhellenic Conference on Informatics*, pages 382–392, 2005.
- [10] Zhou, G. D. Recognizing names in biomedical texts using mutual information independence model and SVM plus sigmoid. *International Journal of Medical Informatics*, 75:456–467, 2006.



High-Recall Gene Mention Recognition by Unification of Multiple Backward Parsing Models

Han-Shen Huang¹

hanshen@iis.sinica.edu.tw

Yu-Shi Lin¹

bathroom@iis.sinica.edu.tw

Kuan-Ting Lin¹

woody@iis.sinica.edu.tw

Cheng-Ju Kuo²

cju.kuo@gmail.com

Yu-Ming Chang¹

porter@iis.sinica.edu.tw

Bo-Hou Yang^{1,3}

ericyang@iis.sinica.edu.tw

I-Fang Chung²

ifchung@ym.edu.tw

Chun-Nan Hsu¹

chunnan@iis.sinica.edu.tw

¹ Institute of Information Science, Academia Sinica, Taipei, Taiwan

² Institute of Bioinformatics, National Yang-Ming University, Taipei, Taiwan

³ Department of Electrical Engineering, Chang-Gung University, TaoYuan, Taiwan

1 Introduction

We considered the gene mention tagging task as a classification problem and applied support vector machines (SVM) to solve it. We selected a large set of features as the input and trained two SVM models with different multiclass extension methods. We found that backward parsing constantly outperformed forward parsing regardless of the multiclass extension methods and obtained high precision rates, but recall rates were not as satisfactory. To enhance recall rates, our approach is to construct divergent but high performance models to cover different aspects of the feature space, and then combine them into an ensemble. We applied union and intersection to combine the outputs of SVM models with that of a CRF model, and successfully enhanced recall rates without degrading too much precision.

2 Method and Results

SVM has been shown to perform well for name entity chunking in the literature (see, e.g., [2, 5]). Name entity chunking is a problem of supervised sequential learning. To apply SVM to this problem, We used a sliding window to convert the problem into a supervised classifier learning problem [1]. We chose five as the width of the window. During the parsing, the information from the two preceding tokens and the two following tokens are used to construct a feature vector for the classifier to assign a class label to the current token. We chose Yet Another Multipurpose Chunk Annotator (YamCha) [2] to build our SVM models because it is tuned for name entity chunking tasks.

We designed our features based on our experience and previous work on name entity recognition [4, 5, 6]. Table 1 shows the set of features. There are 10 feature types with 617,515 feature values in our feature set. As a preprocessing step, we used the GENIA tagger [7] to tokenize sentences and tag part-of-speech (POS) for training and test data. Then we can extract features from the data.

We used an Inside/Outside representation for name entity chunking with *B*, *I*, and *O* class labels. Since SVM is intrinsically a binary classifier, we must extend SVM to handle multiclass problems. We used two popular methods to extend a binary classifier to handle multiclass problems:

- one vs. all: Train a binary classifier for each class against all other classes.

Table 1: Types of features and their possible values

Feature	Value
word	all words in the training data
POS	part-of-speech tagging by GENIA tagger
orthographic	see Table 2 for details
vowel	it is a list of the vowel(s)(a,e,i,o,u) in a word
length	1,2,3,...,5, ≥ 6
morphological I	replacing digits with a “*”(e.g., Abc123→Abc*)
morphological II	replacing each letter and digit with a morphological symbol (e.g., AbcD123→AaaA111)
prefix	1,...,6 gram of the starting letters of the token
suffix	1,...,6 gram of the ending letters of the token
preceding class	class labels(B,I,O) of the two preceding tokens

Table 2: Types of orthographic features and their examples

Feature	Ex.	Feature	Ex.	Feature	Ex.	Feature	Ex.
InitCaps	Abc	SigleDigit	1	BackSlash	/	Apostrophe	,
EndCaps	abc	TwoDigits	12	OpenSqure	[QuotationMark	"
AllCaps	ABC	ThreeDigits	123	CloseSqure]	Greek	α
LowerCase	abc	FourDigits	1234	Colon	:	AminoAcidLong	lysine
WordAndDigits	A1	MoreDigits	12345	SemiColon	;	AminoAcidShort	Lys
InitCapsEndCaps	AbC	Floatpoints	1.2	Percent	%	Nucleoside	Uracil
SingleCap	A	Star	*	OpenParen	(Nucleotide	ATP
TwoCaps	AB	Equal	=	CloseParen)	Roman	V
ThreeCaps	ABC	Plus	+	Comma	,		
MoreCaps	ABCD	Hyphen	-	FullStop	.		

- one vs. one: Train a binary classifier for each pair of classes and select the class appearing in the most outputs.

We also trained a conditional random field (CRF) model to increase the divergence of our ensemble. The CRF model was trained using MALLET [3] with a similar set of features.

We compared two parsing directions: forward and backward. In forward parsing, the tagger reads and tags the input sentences from left to right, while in backward parsing, the tagger reads and tags the input sentences from right to left [2]. Table 3 shows the results of our comparison, where backward parsing performed better than forward parsing for both SVM and CRF models, but there is no evidential difference between the SVM models with different multiclass extensions. For all models, precision is substantially better than recall. These models were trained by 10,000 examples selected at random from the data set provided by the organizer and tested by the remaining 5,000 examples.

Table 3: Performance comparison for different models and parsing directions

Model	Forward	Backward
SVM+One vs.All	P:82.81% R:78.27% F:80.48%	P:86.99% R:75.79 %F:81.01%
SVM+One vs.One	P:82.41% R:78.11% F:80.20%	P:85.49% R:79.25 %F:82.25%
CRF	P:86.52% R:79.44% F:82.83%	P:86.77% R:80.39% F:83.46%

P,R and F denote precision, recall, and f-score, respectively.

Our final step is to determine how to integrate results of the three models mentioned above to enhance recall. Weighted majority vote may be a good idea, but regulating the weights for different models is difficult. Poorly assigned weights may degrade the performance. Instead, we simply applied union and intersection to combine these models. Usually, union can enhance recall because it includes more tagging results from different models, but it also degrades precision. In contrast, intersection can filter out false positives and therefore increase precision, but at the expense of recall. To take advantage of both operations but avoid their pitfalls, we applied intersection to the tagging results of the two SVM models and then union with the tagging results of the CRF model as our ensemble model. Table 4 shows the final test results of this model, as well as the final results of the unions of the CRF model with the two SVM models, as reported by the organizer. The results show that our simple ensemble model remarkably enhanced recall, with all recall results ranked in the top quartile, while precision results dropped slightly, compared with the results in Table 3. All f-score results were ranked in the top quartile, too.

Table 4: Experimental results

Run	Ensemble	Performance
1	M1UM3	P:83.27%(3) R:89.34%(1) F:86.20%(1)
2	M2UM3	P:82.98%(3) R:89.58%(1) F:86.15%(1)
3	(M1∩M2)UM3	P:84.93%(3) R:88.28%(1) F:86.57%(1)

The number in the parentheses is the quartile among 21 participants

References

- [1] Dietterich, T. G. Machine learning for sequential data: A review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 15–30, 2002.
- [2] Kudo, T. and Matsumoto, Y. Chunking with support vector machines. In *Proceeding of Second Meeting of North American Chapter of the Association for Computational Linguistics(NAACL)*, pages 192–199, 2001.
- [3] McCallum, A. K. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, 2002.
- [4] McDonald, R. and Pereira, F. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, 6(S6), 2005.
- [5] Mitsumori, T., Fation, S., Murata, M., Doi, K., and Doi, H. Gene/protein name recognition based on support vector machine using dictionary as features. *BMC Bioinformatics*, 6(S8), 2005.
- [6] Takeuchi, K. and Collier, N. Bio-medical entity extraction using support vector machines. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine*, pages 57–64, 2003.
- [7] Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., and Tsujii, J. Developing a robust part-of-speech tagger for biomedical text. In *Advances in Informatics - 10th Panhellenic Conference on Informatics*, pages 382–392, 2005.



Attribute Analysis in Biomedical Text Classification

Francisco Carrero García¹
francisco.carrero@uem.es

José María Gómez Hidalgo¹
jmgomez@uem.es

Jacinto Mata²
mata@uhu.es

Enrique Puertas¹
enrique.puertas@uem.es

Manuel Maña²
manuel.mana@diesia.uhu.es

¹ Universidad Europea de Madrid, Tajo S/N, 28670 Villaviciosa de Odón, Madrid, Spain
² Universidad de Huelva, Dr. Cantero Cuadrado 6, 21071 Huelva, Spain

Abstract

Text Classification tasks are becoming increasingly popular in the field of Information Access. Being approached as Machine Learning problems, the definition of suitable attributes for each task is approached in an ad-hoc way. We believe that a more principled framework is required, and we present initial insights on attribute engineering for Text Classification, along with a software library that allows experiment definition and fast prototyping of classification systems. The library is currently being used and evaluated in our Information Access projects in the biomedical domain. In this paper we describe how we have used it in the Gene Mention and the Protein-Protein Interaction (Protein Interaction Article) tasks in the Biocreative II Challenge.

Keywords: text classification, machine learning, attribute engineering.

1 Introduction

Text Classification is a subtask of Information Retrieval that involves the automatic labeling of textual elements (such as documents, words or groups of words) with categories or classes. Several classification tasks are used to improved information access to a wide variety of information domains, being biomedicine one of the most prominent ones. Competitions as the Biocreative I and II challenges demonstrate important classification tasks for this domain.

Within the latest years, Machine Learning (ML) techniques have become the main approach to build classification systems. One reason for this is related to the fact that this approach has proved to be very effective for some classification tasks and knowledge domains. However, it should not be forgotten that there is an increasing number of resources available to researchers, and this fact has contributed to reduce the development costs of classification systems with a ML approach.

An example can be found Text Categorization, a subtask of Text Classification in which categories are predefined. Sebastiani [3] states that the advantages of using ML with Text Categorization “are an accuracy comparable to that achieved by human experts”, and “considerable savings in terms of expert manpower, since no intervention from either knowledge engineers or domain experts is needed for the construction of the classifier or for its porting to a different set of categories.”

Domain experts now have become responsible for creating the resources that will be used as data sets. For instance: when working with web pages, web directories such as Yahoo! provide millions of web pages classified by human experts; Ohsmed is a set of references from the on-line medical information database MEDLINE; and Reuters Corpus Volume 1 is currently the most widely used collection for news categorization research.

The selection of attributes that will be used to represent the instances is currently one of the most critical issues in Text Categorization, especially because other stages in the Text Categorization cycle have available libraries

and tools that simplify the process. For instance, WEKA [5] is a suite of ML software developed at the University of Waikato with Java technology, which implements several algorithms from various learning paradigms.

Our work is focused on the analysis of common elements in the multiple Text Classification tasks. We have developed a framework and software library that allow together the analysis, modeling and fast prototyping of classification systems, supporting both the experimentation phase and the development of functional system prototypes.

This library is being used in two R&D projects, Isis and Sinamed [4], whose objective is to enhance Information Access in the medical domain through the improvement and utilization of Text Classification tasks, like Text Categorization, Automated Text Summarization, and Biological Entity Recognition. While Sinamed is focused in basic research, Isis involves dealing with actual patient medical records. We are currently using our library for tasks addressed in these projects. As a part of this research, we have taken part in the Gene Mention and the Protein-Protein Interaction (Protein Interaction Article) tasks in the BioCreative II Challenge, following the approaches we describe in this paper.

2 Organization of Text Classification Tasks

A possible organization of Text Classification tasks can be done on the basis of the granularity of the text elements. Therefore we can either consider words, phrases or documents as atomic elements. Another way to categorize Text Classification is to determine whether the learning process is supervised or unsupervised. These two organizations are often combined to characterize a Text Classification task [2]. Examples of this are the following ones:

- Text Categorization¹ works with documents and uses supervised learning.
- Name Entity Recognition is a task that assigns predefined labels (supervised learning) to sequences of words that represent a certain kind of entity.
- Text Summarization addresses both the problem of selecting the most important portions of text and the problem of generating coherent summaries, using supervised learning.

Focusing on an actual task addressed by our framework, we can distinguish three different phases in the life cycle of a Text Categorization system: document indexing, classifier learning and classifier evaluation [1].

- Document indexing involves mapping a document into a compact representation of its content that can be directly interpreted both by a classifier-building algorithm and by a built classifier. In this stage the decisions on attribute representation and attribute selection become critical.
- Classifier learning. A text classifier is automatically generated by a general inductive process. This process infers the characteristics that any document should have to be classified under each category by observing the characteristics of a set of pre-classified documents
- Classifier evaluation. Different measures to evaluate a classifier are: effectiveness (success rate), training efficiency (required time in average to train a classifier) and classification efficiency (required time in average to classify a document).

There is a number of software libraries that provide support to the latest phases. However, document indexing is most often approached in an ad-hoc fashion. Therefore, we believe that a framework is required to better understand the value of potential representation elements (attributes), not only in text Categorization, but in general, in all the Text Classification tasks.

3 A Taxonomy of Attribute Types for Text Classification

The kind of attributes to use is an important aspect to be considered when experimenting with different ML algorithms. Nevertheless, when computing an attribute given a training instance, other criteria should be taken into account, related to the set of examples that must be processed to set a value for an attribute of a single example. We propose the following types:

- Intrinsic. When computing an attribute for a given example, only information from that example is used. E.g.: the length of a text in Text Categorization.
- Contextual extrinsic. The information is obtained from the processed example, but also from other examples that have a strong relation with it. E.g.: occurrence of a word in a text cited by the current text.
- Global extrinsic. The information comes from all the examples in the set. E.g.: occurrence of a word in the rest of the texts included in the set.

¹ These tasks, along with many others, are fully described in [2].

We can give some representative examples for another Text Classification task: Name Entity Recognition (NER). The main paradigm used to solve this problem involves reducing it to a word tagging problem and face it using ML methods. The idea consists on identifying lexical, morphological, syntactical and orthographic attributes to characterize every word in the text, and applying ML algorithms to categorize the words as part of a named entity or not. Within this task, a common intrinsic attribute could analyze the current word to check if the first character is a capital letter. An example of contextual extrinsic features could be the part of speech for the current word. Finally, a global extrinsic attribute could pre-process all the words in the set to get a list of words likely to be part of an entity, and then check if the current word belongs to this list.

This organization of attributes has strongly motivated the organization of our software library, detailed in the next section.

4 Library Organization

As we have mentioned before, we have developed a framework and software library that allow the analysis, modeling and fast prototyping of classification systems. It runs part of the document indexing process, specifically the mapping of a document into a compact representation, but also helps to build fast functional prototypes to make experiments with different tasks and attribute sets.

With these considerations in mind, we identified the following requirements for the library:

- Simple and flexible attribute definition.
- Simple and flexible definition of input and output formats.
- Easiness to build prototypes for diverse classification tasks.

The library is organized in the following packages:

- jtLib.jtFormatter. This is the core package of the library, and contains the classes that support all the mapping process. The main class is JTFormatter, which implement an abstract mapper by running the following tasks in the given order:
 1. Pre-processing of attributes.
 2. Processing of intrinsic and contextual extrinsic attributes.
 3. Processing of global extrinsic attributes.
 4. Generation of data set in the object representation.
- jtLib.attribs. It contains some common pre-defined attributes to be used in diverse Text Classification tasks. We compute these attributes according to the following characteristics:
 1. The granularity of the text element to be classified.
 2. The internal representation.
 3. Its intrinsic or extrinsic nature.
- jtlib.tasks. It holds several sub-packages that implement different classifiers, like a text categorizer and an entity recognizer

To provide a flexible attribute selection we have developed an attribute configuration system based in XML. It means that a formatter can use any attribute defined in the jtLib.attribute package just by adding it to an XML configuration file. Thus, several parameters can be defined in the file, such as the name of the attribute, the class that will compute it, the size of the attribute window to be considered, the position of the current attribute within the window, the intrinsic or extrinsic nature (needed to decide if an attribute needs a pre-processing phase), etc. This helps to build different attributes vectors easily in a systematic way, therefore allowing to run multiple (and long) experiments to find the best vector with little human intervention.

The following code is a fragment example of an attribute configuration file. We define a contextual extrinsic attribute that computes if the words in a window of two words before to two words after begin with a capital letter.

```
<list>
  <attribute>
    <name type="string">initialCapitalLetter</name>
    <class type="string">jtLib.attribs.InitCaps</class>
    <domain type="string">{ 0, 1 }</domain>
    <size type="int">5</size>
    <prev_window type="int">2</prev_window>
    <next_window type="int">2</next_window>
    <scope type="string">contextual extrinsic</scope>
  </attribute>
</list>
```

The next two sections provide an example of the work we have developed for the 2006 Biocreative challenge.

5 Approach and performance on the Gene Mention Task

In this task we applied a simple process to build the classifier, with the aim to get a first working version with a low effort, and then concentrate on attribute analysis. During the first part of this process (get the working version) we used our JTLib library and the WEKA package for the following stages:

1. Document indexing. We used JTLib to develop an application that processes the training data (the 15,000 sentences preceded by a identifier) to obtain a representation based on the selected attributes and configured into the input WEKA format (ARFF).
2. Dimensionality reduction. Once the former ARFF file is generated, we used WEKA to process it aiming to find the attributes with best information gain. Then, we obtained a new and definitive training file to build the classifier. Table 1 shows the selected attributes classified by the information needed to compute them. We used 28 attributes to characterize each instance. Table 2 collects the ranking of the most relevant attributes obtained from the application of information gain.

Table 1. Selected attributes and their classification

Attribute type	Attribute Name	Description
Intrinsic	hyphen punctuation initCaps lettersAndDigits number	Set of lexical and morphological features depend only on the current word itself: begins with a capital letter, contains letters and digits or only letters or numbers, is a punctuation mark, or includes a hyphen.
Global extrinsic	frequentWords frequentWordsInEntity prev1Unigrams prev2Unigrams startingWords endingWords	These attributes receive a value indicating whether the word can be usually found as a starting or ending word in a named entity, if it is a frequent word inside or outside an entity, or if it can be found just before the beginning of an entity.
Contextual extrinsic	endOfSentence punctuation $\pm n$ initCaps $\pm n$ frequentWords $\pm n$ frequentWordsInEntity $\pm n$	EndOfSentence indicates if the current word is placed at the end of a sentence, so its value is given by the context. The others are values obtained from a window of ± 2 words.

Table 2. Ranking of attributes using information gain

Ranking	Attribute
1	frequentWords
2	frequentWordsInEntity
3	frequentWords - 1
4	frequentWords + 1
5	endingWords
6	frequentWordsInEntity - 1
7	prev1Unigrams
8	frequentWordsInEntity + 1
9	lettersAndDigits
10	startingWords
11	frequentWords - 2
12	frequentWords + 2
13	frequentWordsInEntity - 2
14	prev2Unigrams
15	hyphen

Table 3. Effectiveness on the test data

	C4.5 unpruned	C4.5 pruned
Precision	50.09	53.37
Recall	46.12	42.46
F-measure	48.02	47.29

3. Classifier learning. Using WEKA, we generated a set of models with different Machine Learning algorithms (Naïve Bayes, kNN, AdaBoost with Naïve Bayes). From these classifiers, C4.5 decision tree

achieved the best results. The C4.5 algorithm allows to done a pruned tree in a reduced time but increasing the error rate. We built two classifiers, both pruned and unpruned.

4. Evaluation of text classifiers. Table 3 shows the effectiveness of both classifiers evaluate using the test data set. The C4.5 unpruned achieves a scarcely improvement of the F-measure respect to C4.5 pruned. However, the time needed to build the model of the pruned version is a 22% of the time required by the unpruned version. The classification time of the pruned algorithm is also very lower, being the 6% of the time employed by the C4.5 unpruned.

6 Approach and performance on the PPI-IAS Task

For this task we applied the same process as in Gene Mention task, also using JTLib and WEKA:

1. Document indexing. After processing the 5,500 abstracts that make up the training data, a representation based on the selected attributes is obtained and configured into the ARFF format. In this case, the input data is composed of all texts pre-classified with either curation relevance = 1 or curation relevance = 0.

In order to build a model, we have defined a set of attributes that consists of the most relevant words (unigrams) for classification, as well as the most relevant pairs (bigrams) and trios (trigrams) of words (all of them in lower case). The lists of n-grams are determined using a correlation coefficient, and each n-tuple of words becomes an attribute, with a value of 1, if it is contained in the text, or 0 otherwise.

2. Dimensionality reduction. During the iterative process, we searched for the n-tuples with higher and lower correlation coefficient to build the attribute vector, and tried with several combinations of amounts of unigrams, bigrams and trigrams. The n-tuples with higher coefficient are likely to present higher information gain for documents classified as PPI articles, whereas those tuples with lower coefficient are representative of documents not classified as PPI articles. Table 4 shows some of the n-tuples with higher correlation coefficient. Some of the bigrams and trigrams seem to be too general (i.e.: “we show that”) but others seem to correspond to some experiments with the SUISEKI interaction frames [6].

As a consequence of the ranking of tuples, in our process to obtain the best set of attributes we made some experiments using a stemmer, but the results were unsatisfactory. Moreover, we tried to group some biomedical terms using prefixes of one, two and three letters length, but we did not get a good outcome either.

Table 4. n-tuples with higher correlation coefficient

unigrams	bigrams	trigrams
interaction	two hybrid	yeast two hybrid
domain	domain of	two hybrid system
binding	yeast two	we show that
hybrid	with the	the yeast two
yeast	interacts with	a yeast two
with	in vitro	the interaction of
interacts	the interaction	to interact with
terminal	interact with	in vitro and
complex	interaction between	is required for
interact	interaction with	two hybrid screen

3. Classifier learning. After several experiments with different Machine Learning algorithms, such as Naïve Bayes, C4.5 decision tree and Adaboost, Adaboost with Naïve Bayes showed to be the most effective. Then, we continued our experiments only with the latter, considering different attributes vectors.
 4. Evaluation of text classifiers. Table 5 shows the evaluation results obtained using the test data set. The linear increment on the amount of n-tuples used increases precision and F-Measure, but results in a poorer recall. The increment in the amount of bigrams and trigrams produces a higher recall again, but with lower precision and F-measure. The experiments realized with the training set proved that increasing the number of attributes would produce similar results, but not necessarily better.

Our experiments with these classifiers over the training set also produced a similar recall, but a much higher precision (> 0.95). Whereas cost sensitive learning could be of great help to improve our results, we find positive that recall values are always near to 1.

Table 5. Effectiveness on the test data

	unigrams/bigrams/trigrams		
Precision	3628 / 14240 / 6817	5140 / 18510 / 10224	5140 / 24206 / 18174
Recall	0.555	0.583	0.539
F-measure	0.981	0.952	0.984
	0.709	0.723	0.696

7 Conclusions and Future Work

In this paper, we have described the approaches we have followed in the Gene Mention and the Protein-Protein Interaction (Protein Interaction Article) tasks in the Biocreative II Challenge. The participation of our team in the Biocreative competition has primary served us as a proof-of-concept for our systematic approach to feature engineering in text classification tasks. We believe we have obtained reasonable results with respect to the effort we have invested in the competitions. We intend to join other competitions in order to refine our approach, and to improve the feature modeling library. In particular, we address the next two competitions:

- The International Challenge on Classifying Clinical Free Text Using Natural Language Processing², organized by the Computational Medicine Center (Cincinnati, Ohio, US). The goal of the challenge is to create and train computational intelligence algorithms that automate the assignment of ICD-9-CM codes to clinical free text. This is a kind of Text Categorization task, very related to the goals of our research projects. The challenge is currently running (as by mid February).
- The second edition of the i2b2 Natural Language Processing Challenge³. This challenge is promoted by the Informatics for Integrating Biology and the Bedside Center, an NIH-funded National Center for Biomedical Computing (based at Partners HealthCare System). The first edition has proposed two tasks: the medical record anonymization (a form of Named Entity Recognition, in which entities are after changed to preserve patient privacy), and the Smoking Status detection (a form of Text Categorization, in which a label stating the smoking status of the patient has to be assigned to medical records). We have taken part in this latter task, with average results.

Once we have further tested our library, and we can fully trust it and provide some teaching material, we will release it as opensource software, intended to complement WEKA for text analysis.

In the framework of the project Sinamed, we also plan to build several Information Access systems in biomedicine, specifically targeting the medical doctors when preparing their patient cases and when researching, and for students preparing and documenting assignments. We intend to connect the clinical record information with related scientific information, by using Text Categorization of documents according to the SNOMED ontology concepts. We also have to meet the privacy requirements of patients' information by using the anonymization techniques proposed by others in the i2b2 Challenge.

References

- [1] Avancini, H.; Rauber, A. and Sebastiani, F. *Organizing Digital Libraries by Automated Text Categorization*. Proceedings of ICDL 2004, TERI, 2004, 919 - 931.
- [2] Lewis, D.D.: *Representation and learning in information retrieval*. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, US, 1992.
- [3] Sebastiani, F.: *Machine learning in automated text categorization*. ACM Comp. Surveys 34, 1–47. 2002
- [4] Buenaga, M.; Maña, M.; Gachet, D. and Mata, J. *The SINAMED and ISIS Projects: Applying Text Mining Techniques to Improve Access to a Medical Digital Library*. 10th European Conference, ECDL 2006, Alicante, Spain, September 17-22, 2006. 548-551.
- [5] Witten, I. H. and Frank, E. *Data mining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufman, San Francisco, CA, USA, 2000.
- [6] Blaschke C and Valencia A. *The potential use of SUISEKI as a protein interaction discovery tool*. Genome Informatics Series 12: 123-134. 2001.

² <http://www.computationalmedicine.org/challenge/index.php>.

³ <http://i2b2.org/NLP/>.



Penn/UMass/CHOP Biocreative II systems

Kuzman Ganchev ¹

kuzman@seas.upenn.edu

Koby Crammer ¹

crammer@seas.upenn.edu

Fernando Pereira ¹

pereira@seas.upenn.edu

Gideon Mann ²

gmann@cs.umass.edu

Kedar Bellare ²

kedarb@cs.umass.edu

Andrew McCallum ²

mccallum@cs.umass.edu

Steven Carroll ³

carroll@genome.chop.edu

Yang Jin ³

jin@genome.chop.edu

Peter White ³

white@genome.chop.edu

¹ Department of Computer and Information Science, University of Pennsylvania, Philadelphia PA

² Department of Computer Science, University of Massachusetts, Amherst MA

³ Division of Oncology, The Children's Hospital of Philadelphia, Philadelphia PA

Abstract

Our team participated in the entity tagging and normalization tasks of Biocreative II. For the entity tagging task, we used a k -best MIRA learning algorithm with lexicons and automatically derived word clusters. MIRA accommodates different training loss functions, which allowed us to exploit gene alternatives in training. We also performed a greedy search over feature templates and the development data, achieving a final F-measure of 86.28%. For the normalization task, we proposed a new specialized on-line learning algorithm and applied it for filtering out false positives from a high recall list of candidates. For normalization we received an F-measure of 69.8%.

Keywords: entity tagging, entity normalization, linear sequence models

1 Introduction

We submitted entity tagging and entity normalization systems. For the entity tagging task, we used a rule-based tokenizer followed by a linear sequence model trained using a 5-best MIRA learning algorithm [7]. MIRA accommodates different training loss functions, which allowed us to exploit alternative labelings in training and tune the loss function for higher performance. We also augmented the feature set with curated lexicons and with automatically derived unsupervised word clustering features and found that their combination gives a more than additive gain.

For the normalization task, we used our entity tagging system trained for high-recall and use this to generate a list of mentions for each abstract. We used a simple matching algorithm to find potential gene aliases for each mention, and a new specialized online learning algorithm to filter out false positives from this initial high-recall set of candidates. Some of the features used by the learning algorithm are based on learning what kinds of changes indicate different aliases for the same gene and what kinds indicate different genes.

2 Entity Tagging

Training and test sentences are first tokenized with a rule-based tokenizer. A linear sequence model assigns one of the three B,I, and O tags to each word, as in [12]. We started from a CRF-based system [4] with features similar to those in earlier work [9]. We made three major changes to the previous system:

1. We trained the model with the k -best MIRA algorithm using a loss function that considers alternative labelings and balances precision and recall (Section 2.1), rather than with CRF training [4].
2. We added word features based on distributional clustering (Section 2.3).
3. We performed feature selection by greedy search over feature templates (Section 2.4).

Together, these changes yielded an overall improvement of 4.3% absolute performance improvement (24% relative error reduction) over the baseline system.

2.1 K -best MIRA and Loss Functions

In what follows, x denotes the generic input sentence, $Y(x)$ denotes the set of possible labelings of x , $Y^+(x)$ the set of correct labelings of x . There is also a distinguished “gold” labeling $y(x) \in Y^+(x)$. For each pair of a sentence x and labeling $y \in Y(x)$, we compute a vector-valued feature representation $f(x, y)$. Given a weight vector w , the score $w \cdot f(x, y)$ ranks possible labelings of x , and we denote by $Y_{k,w}(x)$ the set of k top scoring labelings for x . As with hidden Markov models [11], for suitable feature functions f , $Y_{k,w}(x)$ can be computed efficiently by dynamic programming. A linear sequence model is given by a weight vector w .

The learning portion of our method requires finding a weight vector w that scores the correct labelings of the test data higher than incorrect labelings. We used a k -best version of the MIRA algorithm [2, 7, 6]. This is an online learning algorithm that for each training sentence x updates the weight vector w according to the rule:

$$\begin{aligned} w_{\text{new}} &= \arg \min_w \|w - w_{\text{old}}\| \\ \text{s.t. } &\forall y \in Y_{k,w}(x) : w \cdot f(x, y(x)) - w \cdot f(x, y) \geq L(Y^+(x), y) \end{aligned}$$

where $L(Y^+(x), y)$ is a measure of the loss of labeling y with respect to the set of correct labelings $Y^+(x)$.

The most straightforward and most commonly used loss function is a Hamming loss. This sets the loss of labeling y with respect to the gold labeling $y(x)$ as the number of tokens where the two labelings disagree. The Hamming loss does not make use of the alternative labelings provided in Biocreative.

As we show in section 2.5, a better loss function uses a weighted combination of the number of false positive gene mentions and false negative gene mentions in the sentence. This allows the algorithm to prefer labelings $y \in Y^+(x)$ over $y \notin Y^+(x)$. Notice that the update rule still requires that the gold labeling $y(x)$ has to have at least as high a score as any of the proposed labels $y \in Y_{k,w}(x)$. Our experience showed that getting a high precision is relatively easier than a high recall, so we weigh the number of false negatives higher than the number of false positives.

2.2 Lexicons

In our experiments, we used a number of curated lexicons to create lexicon membership features as in previous work [9]. During the greedy feature search we found that removing some of these actually improved performance as did adding some others. The final set of gene and non-gene lists we used were two gene lists developed in previous work [3, 14], a general biology term list [13], a gene list from the Human Genome Organization, a list of chemicals extracted from PubChem and a list of diseases from the MeSH ontology. Lexicons we considered using that did not help included a list of common terms, a list of common gene acronyms and a list of amino acid names (among others).

Feature category	Description of final features
Token	The word at position -2...0
Token Suffix	the last two characters of the words at positions -2...1
Token Prefix	the first 2- and 4- characters of the words at positions -2...1
Token n -grams	any 2-, 3- and 4- consecutive characters of current token
Part of speech (POS)	POS of words at position -1...0 as well as the conjunction of the POS of the words at positions -3...0.
Token and POS	the conjunction of POS and token also at position 0
Cluster Features	identity of the cluster of the words at position -1...1 (see Section 2.3)
Lexicon Features	what lexicons match at positions -2...1 (see Section 2.2)

Table 1: The features used in the final system. Position 0 refers to the token at the position we are considering; negative positions are offsets to the left, positive positions are offsets to the right. With the exception of the cluster features, we distinguish features based on their position, so the feature corresponding to the word at position 0 is different from the word at position -1.

2.3 Distributional Clustering

An 85 million word subset of MEDLINE was used to cluster words by bigram language model perplexity into a binary tree [1]. Different depth tree cuts were then applied to produce 5 clustering features at different levels of granularity for each word type in the tree [10]. Thus, for each word type that has been clustered there are 5 different non-independent cluster features generated by the clustering. These additional features are then added the feature function in training and testing. On our development data, adding these features produced a 0.7% improvement in the best system and as much as 1.3% improvement in inferior systems.

2.4 Greedy Search

For the baseline system, we started from a feature set similar to that of our previous work [9] to which we applied feature selection as follows. Features were grouped by feature templates. For example, there are many features for the identity of the current token (one for each token type), but we group all of these into a single identity feature template. Starting with our initial list of feature templates, we repeatedly remove the one whose removal results in the greatest increase in the score of the development data, until no further improvement is possible. Removing just one feature template in this way requires training one model for each removal candidate. Once we cannot improve development data performance, we start adding feature templates from a list of candidates. This resulted in some unexpected additions and non-additions. For example, we found that adding a conjunction of four POS tags helps performance, while adding our list of gene acronyms actually hurts performance. Table 1 describes the features used in the final submission.

Even though there are hundreds of thousands of features, there are only dozens of feature templates, so doing this optimization on the development data does not lead to very severe overfitting: the F-score of the final system on the development data was within 1% of that on unseen data. The initial performance of the baseline system is similar to that of the “public access” system described in the following section. Despite the fact that the baseline system had access to some lexicons, it had poorer feature selection overall.

2.5 Analysis and Results

In development, we split available labeled data into training, development and test sets (80%, 10% and 10% respectively). We did not use test data during development, so we could use it to compare a few approaches (Table 2). The baseline system contains some lexicons but did not use the greedy search to

Method Features	CRF			MIRA Hamming			MIRA FP+2FN		
	P	R	F-1	P	R	F-1	P	R	F-1
Baseline	84.3	80.0	82.1	85.3	78.5	81.7	83.2	83.9	83.5
Public access	83.4	79.1	81.2	85.5	78.3	81.8	83.0	86.3	84.6
Clusters	84.6	80.6	82.6	85.7	79.0	82.2	83.1	86.9	85.0
Lexicons	85.0	80.1	82.5	86.7	80.1	83.3	84.4	86.9	85.7
All Resources	85.9	81.3	83.5	87.4	81.6	84.4	85.1	87.7	86.4

Table 2: Precision, recall, and F-measure on held-out data. CRF is a standard CRF trained with prior variance of 10. MIRA Hamming is 5-best MIRA trained with Hamming loss. Mira FP+2FN is 5-best MIRA with a loss counting false positives and twice false negatives. The rows correspond to different feature sets: just features from the data provided, adding word clusters only, adding lexicons only, and all features (all features subject to feature selection). The best system used all the methods for a 4.3% absolute improvement (24% error reduction) over the baseline system.

optimize feature templates, while the public access system did use the greedy search. Clusters, lexicons and all resources describe systems that add to the public access system. These results show that out of the three improvements, the use of MIRA with a tuned loss function and using the alternative labelings yielded the highest performance improvement over the baseline CRF system (3%). Use of alternative labelings and balancing of false positives and false negatives in the loss function yielded around 2-3% performance improvement. Both the automatic cluster features and the curated lexicons gave around 1% F-measure improvement over the baseline. In conjunction with MIRA with the tuned loss function, the lexicons still provide around a 1% improvement over the baseline MIRA system, while the the automatic clusters provide only around a .5% improvement. Surprisingly, in the final system, adding both lexicons and clusters yielded a 1.8% improvement.

3 Gene Normalization

Our approach is in some ways similar to that of [8]. Like them, we first generate a high-recall list of candidate mentions with corresponding aliases and gene IDs, and then use a linear classifier to filter out the false positives. Our system differs in the way that we find an initial list of candidates (Section 3.1,) the learning algorithm (Section 3.3) and in the features that we used to make the decisions (Section 3.2).

3.1 Gene Mentions and Matching

We used the gene tagger described in Section 2. Since our approach for the gene mention task will filter out incorrect gene mentions later, we used a loss function to maximize recall: the loss of a labeling for a sentence was set to the number of false negatives (with respect to the true labeling). This resulted in a recall on the gene mention tagging task of a little over 90%.¹

For each gene mention in the high-recall list, we return the list of gene ID/alias pairs where the alias is the same as the text of the mention after normalization steps that include removal of common words (such as “gene”, “protein”, “mouse”), replacement of digits with the corresponding roman numerals, removal of spaces and dashes, and case conversion. This yields an initial candidate list that has a recall of 77.2% but a precision of only 37.3%. We tried more aggressive matching rules, but we found that this makes the next learning stage too difficult for the small amount of available training data.

¹We cannot estimate this exactly because we used all available labeled data to train the model.

3.2 Filtering Features

To avoid overfitting due to the small training set, we used only 89 features, including the number of candidates competing for the same mention, the number of candidates that agree on the gene ID, and which of “human”, “rat” and “mouse” appears closest to the mention. A set of more complicated but very useful features are based on a learned string string alignment model [5].

The string edit distance model is trained to maximize the probability alignments between aliases of the same gene, while minimizing the probability of alignments between aliases of different genes. For each candidate, the model gives a probability that the gene mention to the alias, which is converted into the following features: the binned value of the probability, the rank of the current candidate among all candidates in the abstract, and the rank of the current candidate among candidates for the same mention. These features resulted in a significant improvement (about 2% in F-measure) in our cross-validation development runs.

3.3 Learning Algorithm

Our model learns to distinguish a candidate (mention-alias pair) containing a true mention of a gene from a false positive. Unfortunately, the training data is incomplete: for each abstract we have only a list of gene IDs. To overcome this problem we created a MIRA-inspired (Section 2.1) online learning algorithm that makes use of the correct gene IDs (given to us), as well as its current predictions to figure out which candidates to require be correct.

More formally, let C be the set of candidates for a particular abstract, C_{top} be the set of current highest scoring candidates for each correct gene ID, and C_{FP} be the set of candidates that correspond to an incorrect gene ID, but were scored above some threshold θ by the current model. Then, our algorithm performs the following update

$$\begin{aligned} w_{\text{new}} &= \arg \min_w \|w - w_{\text{old}}\| + \gamma \max_c \xi_c \\ \text{s.t. } & w \cdot c \geq 1 - \xi_c && \forall c \in C_{\text{top}} \\ & -w \cdot c \geq 1 - \xi_c && \forall c \in C_{\text{FP}} \\ & \xi_c \geq 0 && \forall c \in C_{\text{FP}} \cup C_{\text{top}}. \end{aligned}$$

The ξ_c are slack variables to account for non-separable data. For our experiments, we used a θ of 0.2 and a γ in the range $0.005 \leq \gamma \leq 0.5$.

4 Acknowledgements

This work was supported in part by the US National Science Foundation under ITR grant EIA-0205448, in part by the Center for Intelligent Information Retrieval, in part by DoD contract #HM1582-06-1-2013, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, and in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0427594. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

- [1] Peter Brown, Peter deSouza, Robert Mercer, Vincent Della Pietra, and Jenifer Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.
- [2] Koby Crammer. *Online Learning of Complex Categorial Problems*. PhD thesis, Hebrew University of Jerusalem, 2004.

- [3] Haw-ren Fang, Kevin Murphy, Yang Jin, Jessica Kim, and Peter White. Human gene name normalization using text matching with automatically extracted synonym dictionaries. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 41–48, New York, New York, June 2006. Association for Computational Linguistics.
- [4] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289, 2001.
- [5] Andrew McCallum, Kedar Bellare, and Fernando Pereira. A conditional random field for discriminatively-trained finite-state string edit distance. In *Conference on Uncertainty in AI (UAI)*, 2005.
- [6] Ryan McDonald, Koby Crammer, and Fernando Pereira. Flexible text segmentation with structured multilabel classification. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 987–994, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.
- [7] Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 91–98, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.
- [8] Ryan McDonald, Jay Crim, and Fernando Pereira. Automatically annotating documents with normalized gene lists. In *BMC Bioinformatics*, 2005.
- [9] Ryan McDonald and Fernando Pereira. Identifying gene and protein mentions in text using conditional random fields. In *BMC Bioinformatics*, 2005.
- [10] Scott Miller, Jethran Guinness, and Alex Zamanian. Name tagging with word clusters and discriminative training. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 337–342, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.
- [11] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):275–285, 1989.
- [12] Lance Ramshaw and Mitch Marcus. Text chunking using transformation-based learning. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, Somerset, New Jersey, 1995. Association for Computational Linguistics.
- [13] Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in full text articles. In *Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain*, pages 9–13, Philadelphia, July 2002. Association for Computational Linguistics.
- [14] Lorraine K. Tanabe and W. John Wilbur. Generation of a large gene/protein lexicon by morphological pattern analysis. *J. Bioinformatics and Computational Biology*, 1(4):611–626, 2004.



Text Detective: Gene/protein annotation tool by Alma Bioinformatics

Rafael Torres¹

torres@bioalma.com

Leandro Pascual¹

pascual@bioalma.com

Pablo D. Sánchez¹

sanchez@bioalma.com

Christian Blaschke^{1,2}

blaschke@bioalma.com

¹ Alma Bioinformatics SL, Ronda de Poniente, 4 2 C-D
E-28760 Tres Cantos, Madrid, Spain

² To whom correspondence should be addressed.

Abstract

Text Detective is a system based on carefully constructed rules and lexicons that detects genes and proteins mentioned in biomedical abstracts. Text Detective was used in the Gene Mention (GM) and Gene Normalization (GN) tasks during BioCreative 2007 . The results were as follows: P=84.3; R=68.6; F=75.6 (quartile 3 - Q3, quartile 4 - Q4) for GM and P=74.3; R=80.7 ; F=77.4 (quartile 1 - Q1) for GN.

Keywords: named entity recognition, BioCreative, gene name disambiguation.

1 Introduction

Text Detective [1] (TD) detects genes and proteins mentioned in biomedical abstracts using specifically designed rules and lexicons. We participated in the Gene Mention (GM) and Gene Normalization (GN) tasks during BioCreative 2007. The best results were as follows: P=84.3; R=68.6; F=75.6 (Q3,Q4) for GM and P=74.3; R=80.7; F=77.4 (Q1) for GN.

This article describes the basic features of the TD methods for detecting genes and assigning Entrez Gene IDs to them. We present the results with some examples to discuss the causes of false positive (FP) and false negative (FN) detections.

2 Methods

The system considers that each reference to a gene or protein belongs to one of these categories:

- Functional annotations: a full name that describes the functionality of the gene/protein, e.g. 'thyrotropin releasing hormone receptor'.
- Symbols: an acronym or abbreviation used as a name, e.g. TRHR.

The methods are different according to the category of the name. In the first case, the morphology and the semantics of the set of words included in the functional annotation is highly indicative. In the second case, as the lexical aspects are frequently irrelevant, the system uses the contextual information (the adjoining words that are related to genes and proteins) to detect parts of the text that refer to a gene or protein.

The rules used by the system were all human generated after careful processing and a study of the Medline corpus (see for example section 2.3). The lexicons are extracted from diverse sources (chemical or genomic databases) or by statistically comparing biological and non-biological corpora of documents (e. g. when extracting lists of words that frequently appear in the same context as the gene named in the texts and lists of words that are related to biomedical sciences).

In spite of having different sets of rules, the processes for extracting functional annotations and symbols follow a similar workflow that involves the following steps:

2.1 Sentence extraction

If the text is not presented in the form of sentences, it is first divided into sentences. Each sentence is tokenized and the positions of each token are stored. Each token undergoes a stemming process that removes some punctuation marks and plural word suffixes. These "cleaned" tokens are used in the subsequent stages. The original words and their positions in the document are also stored.

2.2 Tagging the tokens

Each token is tagged with one of the following class names by means of a number of rules and lists:

- *Keyword*: words that are relevant biologically and that indicate an essential feature of the gene/protein, e.g. 'channel', 'receptor'.
- *Stop word*: words that are very frequent in the corpus and with no biological relevance.
- *Location*: biological locations, e.g. 'membrane', 'liver'.
- *Type*: words that are part of a gene/protein name and that serve to distinguish between similar names. This category includes numbers, combinations of letters and numbers, Greek letters, Roman numerals, etc. This class includes the gene symbols, e.g. 'TNFalpha'.
- *Accessory*: words with a relatively low informative content that are found close to the name of the genes/proteins, e.g. 'family', 'subunit'.
- *Bioword*: words with biological meaning.
- *Verb*: a list of predefined verbs.
- *Unknown*: all other uncategorized words.

This categorization is not a POS tagging. Its only purpose is to determine the function for each token in a possible reference to a gene/protein in the document.

2.3 Name chunking and evaluation of the risk

Simultaneously to token labeling, the system constructs strings of contiguous labeled tokens. The system detects the ending of a string when a comma, a stop word, etc. is found. Rules are applied to decide whether a string of tokens corresponds to a gene/protein. These rules are based on the sequence of classes that the consecutive tokens of a string are labeled with, and are different for functional annotations and for symbols. For example, in the case of functional annotations, the string must contain at least one token belonging to the 'keyword' class.

In the case of symbols, the string must only include tokens of the 'type' class. Additionally, the context of the chain which is supposedly a symbol is also evaluated. We use a pre-calculated file with a list containing the most significant words in the proximity of the gene symbols. The words which are closer to the candidate symbol are scored according to this list. The result of all the scores determines

the acceptance or rejection of the candidate symbol (the final value must exceed a certain threshold value to be considered a valid reference to a gene).

Additionally, the system uses a list that contains a numerical value, the risk factor, associated to each gene symbol. The assignment of certain symbols, such as SCT, to a gene (secretin) has a high risk of being incorrect due to the ambiguity of the term (it can stand also for "stem cell transplant" and others), whilst others, like CYP11B2, has a much lower risk. The risk factor is obtained by analyzing Medline to establish how often the symbol appears in a gene-like context compared to its frequency in a non-gene context.

Both subsets of possible references to genes, those that are functional annotations and those that are gene symbols, together represent the GM task result.

2.4 Gene name normalization (only applied in the GN task)

In this stage, our own dictionary was used (similar to that provided by the BioCreative organizers) which contains the entries from both Entrez Gene and Uniprot for human genes, and attempts to assign the strings that were detected in the previous steps to one dictionary entry.

Once again, there are differences when treating functional annotations or gene symbols:

- In the case of functional annotations, a set of rules are applied to select the definitions in the dictionary of genes that match the annotations found in the text.
 - For symbols, allocating a candidate symbol begins with the selection of all the entries in the dictionary that contain this symbol.

Where there is more than one possible candidate, a disambiguation process is performed to select the most suitable entry. This process uses a list that contains the words for each gene that are relevant when found in the context of this specific gene. Each word is allocated a weighting that represents its importance as a 'context word'. These words are extracted from the functional annotations, GeneRIFs and summary sections in Entrez Gene and Uniprot.

For each reference to a candidate, the system obtains the sum of the context words weightings included in the mentioned file that also appear in the text. The candidate which is finally selected is the one with the highest value, however the detection is only considered to be correct if the value exceeds the minimum risk value associated by the risk factor (see section 2.3). The acceptance or rejection of symbols induces changes in the evaluation parameters and, therefore, the search process is performed iteratively to look for more possible symbols.

3 Analysis

Three runs for the GM task and two runs for GN using different sets of parameters were submitted (see section 3.3). These parameters control the trade-off between precision and recall in the annotations. In some cases, precision is favored by the set of values chosen for the parameters and, in other cases, the recall is improved. Our best results were: $P=84.3$; $R=68.6$; $F=75.6$ (quartile 3, quartile 4) for GM and $P=74.3$; $R=80.7$; $F=77.4$ (quartile 1) for GN.

The following section discusses the main reasons for the false negatives and false positives in each task, giving some examples that were analyzed and considered to be controversial in the annotation sets. In some cases, the discrepancy is due to differences in the definition of what is considered to be a gene or protein in our system.

3.1 Gene Mention Task

3.1.1 False negatives (FN)

There are three main origins for these errors:

Boundaries: In some cases, there are annotations that exceed the limits of the gold standard. The minority of these cases are due to problems with the punctuation (e.g. 'TNF')¹). Sometimes the "excess" belongs to a completely different biological entity (e.g. <activating transcription factor/cyclic AMP>¹). In these cases, the correct meaning of the annotation is distorted or confused. In other cases, this excess adds some interesting feature to the annotated object (e.g. <activated heteromeric N-methyl-D-aspartate receptor channels>). A controversial point is that there are a significant number of cases related to enumeration, e.g. '<transcription factors IRF3>' and '<IRF3>', where our annotation, 'transcription factors IRF3', was considered incorrect, probably due to being plural, with the "correct" answer being 'IRF3'. However, there are some examples where very similar cases were classified as correct ('<Jun> and <Fos family of transcription factors>').

There are also annotations that are "defective" in respect to the gold standard. These defects sometimes severely reduce the significance of the annotation (transcription <factor 4>). Nevertheless, there are several cases where the annotation is sufficiently specific to conserve all its significance ('<HNF3 beta> proteins').

Species: We estimate that recall is reduced a 10-15% when processing documents not related to human. Although the system has been adapted to manage other species, the rules and lexicons are focused towards the human context. For example, genes of bacteria and plants such as 'EcoRI' or 'lox' are not detected. In the majority of cases where the annotation itself contains a name of a species, the system only produces the correct annotation when it contains a symbol that is similar to a human gene, or when a significant number of keywords appear in the context. For example, the system fails to detect gene symbols like 'AtXPO1' but correctly annotates 'ATB2 bzip transcription factor' (At=Arabidopsis thaliana).

'Fuzzy' gene references: TD considers some keywords strongly suggest a reference to a gene or protein: gene, mrna, cdna, peptide, protein, etc. However, the GENETAG corpus [2] has a wider concept of 'gene' and includes open reading frames, (binding) sites, elements, regulons, regions, fragments, repeats, etc. As in the case of species, TD annotates correctly when the portion of text includes a symbol-like reference or sufficient keywords. For example, 'NADPH binding motif' is not detected, while 'Gbeta-like WD-repeat protein' is annotated.

Others: TD discards references to one-letter genes (19 in the test set). Additionally, there are cases where punctuation is not well-resolved (e.g. "AP-1/c-jun family", "seven 'helicase' domains").

3.1.2 False positives (FP)

In addition to errors related to the boundaries of the annotations (40% of the FPs) explained above, the following specific types of errors were detected:

Generic references: This category accounts for about 30% of the FPs. Some of them are in fact very unspecific/low-informative, e.g. 'nuclear factor' or 'protein gene families', however, in other cases we consider that they provide additional chemical or functional information that could be interesting for users, as in 'cytokine inducible nuclear protein' or 'DTPA coupled antibodies'.

Entities belonging to other biological categories: 25% of the FPs. The majority are chemicals including amino acids ('miaserin 30', 'Y701'); procedures/techniques ('T-bil', 'TEM'); species, breeds, strains, etc. ('tsO23', 'Norin') or genomic regions/DNA sequences ('PEST', 'GGTCA/GnnnnAGACC').

Others: 5% of the FPs. They include problems such as incorrect unit of measure annotations (e.g. '34-mer', '500-1000 KB').

3.2 Gene Normalization Task

3.2.1 False negatives (FN)

Three categories of FN were identifiable:

¹The TD annotations are shown between '<' and '>' symbols

Partial annotations (approx. 60% of the cases): TD identifies a chunk of text referring to a gene, but it 'decides' not to assign to any Entrez Gene ID because it does not have sufficient confidence in the allocation. Approximately half of the errors in this category are due to some kind of problem when dealing with punctuation marks (e.g. AIP3/WWP3, huntington-interacting protein). However, the main reason in a large number of cases is that no exact synonym for the reference is listed in Entrez Gene, and there is a reasonable doubt about the assignment. For example, 'TGF-beta' is not assigned ID 7040 ('TGF-beta 1') because this is not a synonym for the ID and additional genes, which are also good candidates, are closely related ('TGF-beta-2', 'TGF-beta-3', etc.).

Word sense disambiguation (WSD, about 8%): The application detects the gene but produces an assignment which is different to that in the gold standard. Some of these cases are difficult to resolve correctly, when based only on the information contained in the abstract, because the 'candidates' have very similar characteristics (consequently, the scoring values that serve to decide between them are also similar). For example, in the gold standard 'Ubc4' in PMID 10531035 is assigned ID 7325 ('ubiquitin carrier protein E2'), but it could equally be assigned ID 7322 ('ubiquitin carrier protein D2').

It seems logical to associate the synonym 'UGTB11' in PMID 8333863 to Entrez Gene ID 10720 ('UDP glucuronosyltransferase 2 family, polypeptide B11'). This is also the correct answer in the gold standard. But 'UGTB11' is also a synonym for Entrez Gene ID 7363, the 'UDP glucuronosyltransferase 2 family, polypeptide B4'. Functionally the two proteins are closely related to each other, given that they form part of the same family of proteins, so the assignment becomes a difficult task.

Another difficult case is the reference to 'two mammalian TR isozymes (TR2 and TR3)' in PMID 10455115. TD annotated Entrez Gene ID 114112 for TR2, and Entrez Gene ID 10587 for TR3, however, in the gold standard set both annotations belong to Entrez Gene ID 10587. This ID contains the synonyms 'thioredoxin reductase 2' and 'TR3' (but not 'TR2'), and ID 114112 contains the synonyms 'thioredoxin reductase 3' and 'TR2' (and not 'TR3'). This case also demonstrates the fact that Entrez Gene contains a certain degree of inconsistency in some of its entries.

A final case of WSD is related to the association 'description (symbol)'. In PMID 10458166 there is a reference to 'bone morphogenetic protein (BMP)' for which the gold standard assigns both the description and symbol to Entrez Gene ID 649 (BMP-1). Nevertheless, there are other possibilities such as IDs 650 (BMP-2), 651 (BMP-3), etc. and nothing in the abstract seems to indicate that BMP-1 is the gene referred by the author.

Lack of detection (approx. 32%): The system does not detect genes due to the presence of some undefined prefixes in the symbol (e.g. 'hRPC62'), the complexity of the gene name (e.g. 'protein homologous to the product of the C. elegans gene lin-7') or the lack of synonyms in Entrez Gene (e.g. 'VP16' only appears in Entrez Gene ID 3054 with the synonym 'VP16 accessory protein').

3.2.2 False positives (FP)

There are a number of categories where these cases can be included (apart from those previously mentioned):

Errors derived from matching 'artificial' synonyms (approx. 30%): TD selects chunks of texts that match a synonym reasonably well. For example, the synonym 'sodium dependent phosphate transporter 1' (Entrez Gene ID 6574) is very similar to the reference 'a type 1 sodium phosphate transporter' in PMID 9149941, however, the assignment is, in fact, incorrect.

Errors derived from the presence of too many generic synonyms in Entrez Gene: These synonyms, when found in the text, are frequently assigned. E.g., 'myosin' is a excessively broad synonym which Entrez Gene associates to ID 389031. Others examples are 'thioredoxin reductase' and 'ubiquitin ligases'.

Sometimes, although the mention is in some way generic, TD assigns it to a gene and this assignment seems to have a high probability to be correct (solely taking into account the information provided by the abstract). For example, TD has associated 'eIF4G' in PMID 9548260 to ID 1981 (eIF4G1) and it

seems this is correct given that it is the commonest member of the complex described in the paper.

References to domains named after a particular gene/protein: Differently to our criteria, GM annotation guidelines point out that these cases do not refer to a gene, e.g. '<pleckstrin> homology domain'.

Genes belonging to a different species: This occurs with orthologues and when the document in turn deals with both human and non-human genes. e.g. '(mouse) snk'. In some cases, it is difficult, even by manual inspection, to determine the association of a gene to one species or another, such as 'Jagged1' in PMID 9315665.

Not a gene: ER ('endoplasmatic reticulum') or ASMD ('anterior segment mesenchymal dysgenesis'). In these cases, a specific method to detect and resolve the acronyms is expected to reduce the incidence of this problem.

3.3 Adjustment of system parameters

The system possesses a set of different parameters that can be modified in order to adjust the recall vs. precision balance in both GM and GN tasks. In diverse runs, the values of some of these parameters were modified after studying their incidence in the training set results. According to the task that is affected, these parameters were:

- For the GM task, we can modulate two elements:
 - The importance given to the risk factor (see section 2.3). We can vary some parameters that establish a minimum value of risk for all the candidate gene mentions that are evaluated. By increasing this value, the importance given to this evaluation based on the risk factor can be increased, and therefore the precision is enhanced, or we can decrease this value, if we want recall taking priority over precision.
 - Another aspect that we can control is the number of words that we analyze as "context words" on the proximity of any candidate to gene mention. When this "window" is small (formed by few words), the precision grows, because we are more strict expecting to find a suitable content in a set of few context words, and therefore the recall is lower. On the contrary, a more extensive window can produce false positives (lower precision), because there are words in the proximity that are incorrectly considered as context words of the candidate.
- For the GN task, the modulation of the two previous aspects has incidence in the result. One additional factor specific for this task can be altered related to the file containing the context words and their associated weightings described in section 2.4. We can modify the threshold value that the sum of the context words weightings needs to exceed in order to assign a particular entry of the dictionary for a gene mention. When this value is high, precision is enhanced but recall decreased and, on the contrary, when the value is low, recall increases and precision decreases.

References

- [1] Tamames,J., Text Detective: a rule-based system for gene annotation in biomedical texts, *BMC Bioinformatics* 2005, 6(Suppl 1):S10, 2005.
- [2] Tanabe, L.; Natalie, X.; Lynne H. T.; Wayne, M.; Wilbur, J., GENETAG: a tagged corpus for gene/protein named entity recognition, *BMC Bioinformatics* 6(Suppl. 1):S3, 2005.



Peregrine: Lightweight gene name normalization by dictionary lookup

Martijn J. Schuemie¹
m.schuemie@erasmusmc.nl

Rob Jelier¹
r.jelier@erasmusmc.nl

Jan A. Kors¹
j.kors@erasmusmc.nl

¹ Biosemantics Group, Medical Informatics Department, ErasmusMC University Medical Center Rotterdam, 's-Gravendijkwal 230, 3015 CE Rotterdam, The Netherlands

Abstract

To achieve high speed with minimal effort, we created a system dubbed Peregrine that performs gene name normalization by simple dictionary lookup followed by several post-processing steps.

Keywords: gene name normalization, dictionary

1 Introduction

For molecular biologists to be able to cope with the massive amounts of information stored in scientific literature, it is not sufficient to simply have an efficient document retrieval system. For instance, to interpret a list of hundreds of up and down regulated genes in a high-throughput experiment, the required information is stored in thousands of relevant articles, too much to read. What is needed is a system that can distill the information from the literature and represent it in a compressed form.

One such system is the Anni tool, developed by the Biosemantics group (www.biosemantics.org/anni). This tool can be used for gene list annotation and knowledge discovery, and has already been applied to current biomedical problems [1].

For tools like these, it is necessary to uniquely identify gene and protein names in literature, and relate these to specific entries in molecular databases. Because the amount of literature that needs to be analyzed is large (Medline alone counts over 16 million records), the method for gene name normalization should be able to analyze large corpora in a reasonable amount of time.

We have therefore chosen to use a lightweight system we named Peregrine, which simply looks up word sequences in a dictionary that is automatically constructed from gene and protein databases. Several post processing steps are applied to reduce the number of false positives and false negatives.

The system is based on a previously published study on gene name normalization [2].

2 Methods

2.1 Tagging system

The Peregrine system translates the terms in the dictionary into sequences of tokens (i.e. sequences of words). When such a sequence of tokens is found in a document, the term, and thus the gene or protein associated with that term, is recognized in the text. Some tokens are completely ignored, since these are considered to be non-informative ("of", "the", "and", and "in"). If the term is considered a 'long form' (i.e. it contains a space and is longer than six characters), the tokens in the thesaurus and in the text are first reduced to their

stem using the NLM Lexical Variant Generator program [3], to allow for small lexical variations.

2.2 Dictionary

We tested the system using two different dictionaries:

1. The dictionary provided by the BioCreAtIvE 2 organization, with 32,975 genes, and 182,989 (non-unique) gene names
2. Our own dictionary, constructed by combining five gene databases [2, 4], with 26,560 genes and 161,928 (non-unique) gene names

2.3 Manual filter

We tested the system on a random selection of 100,000 Medline records. We manually reviewed the 250 most frequently found terms, since these are most likely to be erroneous or highly ambiguous terms. We removed terms that are not really names of genes (e.g. “alternative splicing”, “open reading frame”, and “human”), or are extremely ambiguous (e.g. “CA2”, “obesity”, and “factor 1”). We removed 159 such terms from the BioCreAtIvE dictionary, 98 from our own combined dictionary.

2.4 Spelling variations

To allow for spelling variations not included in the dictionary, we applied two rules to generate new synonyms based on existing terms, as shown to be effective in a previous study [2]:

1. Arabic numbers are replaced with roman numerals and vice versa.
2. If the last part of a gene symbol consists of numbers, a word-delimiter (i.e. a hyphen or a space) is inserted. For example, “ABC1” becomes “ABC-1”. If a word delimiter is present, it is removed. (e.g. “DEF-1” becomes “DEF1”)

2.5 Automatic filter

To remove highly ambiguous terms, especially those that could have been created by the previously mentioned spelling variation generation rules, we applied an automatic filter; We removed terms that consist only of tokens that are either (a) shorter than 3 characters, (b) consist only of numbers or roman numerals, or (c) belong to a set of stopwords. Examples of terms that were removed are: “G 4”, “2.19”, and “And-1”.

2.6 Family name filter

Some gene synonyms in the dictionary are actually family names and should therefore be removed. We used an automatic procedure to identify family names: if a term is also found in the dictionary followed by a number, roman numeral or greek letter, we considered it to be a family name. For instance, “Zinc finger protein” is also detected as a substring in “Zinc finger protein 51”, and is therefore removed as a synonym.

2.7 Simple disambiguation

Similar to Koike et al. [5], we used several simple rules to detect and possibly resolve ambiguous terms:

1. We first determined whether a term is *ambiguous*. A term is considered ambiguous if it *refers to more than one gene in the dictionary*, or when it is *shorter than six characters and does not contain a number*. A non-ambiguous term will automatically be assigned
2. An ambiguous term will only be assigned if a *synonym is found* in the same document, or the *term is the ‘preferred name’ of the gene*.

2.8 Keyword detection

Because the simple disambiguation is rather strict, we also allowed ambiguous terms to be assigned if a *keyword* was found in the same document. A keyword is a word (i.e. a token) that occurs in any of the long-form names of the gene, and appears less than n times in the dictionary as a whole. We have achieved the best results with n = 1,000. For instance, in the term “Prostate Specific Antigen” the word “Prostate” appears less than 1,000 times in the dictionary. If the ambiguous synonym “PSA” is encountered in text, and the word “Prostate” is also encountered, the gene name is recognized.

3 Results

Table 1 shows the precision and recall scores of the system on the BioCreAtIvE 2 test set, after progressive inclusion of the post-processing steps for the two different dictionaries. The highest scores for both dictionaries fall within the second quartile of scores of the BioCreAtIvE 2 competition.

We also tested the speed of the Peregrine system by analyzing a random set of 100,000 Medline records. On a Dual AMD Opteron 248 system, the tagging process and post-processing steps required 213 seconds (about 3.5 minutes).

4 Discussions

The initial difference in precision between the BioCreAtIvE dictionary and our own combined dictionary appears to be primarily caused by additional highly ambiguous terms in the BioCreAtIvE dictionary. Particularly, the term ‘human’ was found as a synonym of 15 genes!

Without extra steps, simple dictionary lookup of (sequences of) words in text leads to a very low precision. Several post-processing steps can be used to boost performance. Especially a set of simple disambiguation rules provide a major increase in precision, but at a loss of recall. Most of the steps described here require little or no manual effort. The resulting system is fast and robust, and can easily be applied to large corpora.

References

- [1] R. Jelier, G. Jenster, L. C. Dorssers, B. J. Wouters, P. J. Hendriksen, B. Mons, R. Delwel, and J. A. Kors, "Text-derived concept profiles support assessment of DNA microarray data for acute myeloid leukemia and for androgen receptor stimulation," *BMC Bioinformatics*, vol. 8, pp. 14, 2007.
- [2] M. J. Schuemie, B. Mons, M. Weeber, and J. A. Kors, "Evaluation of techniques for increasing recall in a dictionary approach to gene and protein name identification," *Journal of Biomedical Informatics*, in press.
- [3] A. McCray, S. Srinivasan, and A. Browne, "Lexical Methods for Managing Variation in Biomedical Terminologies," presented at Proceedings of the 18th Annual Symposium on Computer Applications in Medical Care, 1994.
- [4] J. A. Kors, M. J. Schuemie, B. J. A. Schijvenaars, M. Weeber, and B. Mons, "Combination of genetic databases for improving identification of genes and proteins in text," presented at Proceedings of BioLINK, <http://www.cs.queensu.ca/biolink05/presentations/Kors.pdf>, 2005.
- [5] A. Koike and T. Takagi, "Gene/protein/family name recognition in biomedical literature," presented at BioLINK 2004: Linking Biological Literature, Ontologies, and Databases, 2004.

	BioCreAtIvE dictionary		Combined dictionary	
	P	R	P	R
Tagging system	0.09	0.82	0.42	0.81
+ Manual filter	0.17	0.82	0.44	0.81
+ Spelling variations	0.18	0.84	0.43	0.83
+ Automatic filter	0.36	0.83	0.52	0.82
+ Family name filter	0.48	0.82	0.53	0.82
+ Simple disambiguation	0.77	0.65	0.79	0.68
+ Keyword detection	0.72	0.75	0.75	0.76

Table 1: Precision (P) and Recall (R) for the basic tagging system and the accumulative set of post-processing steps.



Gene Mention and Gene Normalization Based on Machine Learning and Online Resources

Hongfang Liu¹, Manabu Torii¹, Zhang-Zhi Hu², Cathy Wu²
^{{hl224, mt352, zh9, wuc}@georgetown.edu}

¹Department of Biostatistics, Bioinformatics, and Biomathematics, Georgetown University Medical Center, 4000 Reservoir Road NW, Washington, DC, 20007, USA

²Protein Information Resource, Georgetown University Medical Center, 3300 Whitehaven Street NW, Washington DC, 20007, USA

Abstract

The identification of phrases in text representing genes/proteins and the mapping of those phrases with entries in databases are critical for literature mining applications in the biomedical domain. In this paper, we report the participation of BioTagger, an automated gene/protein name identification and normalization system, in both the gene mention and the gene normalization tasks of BioCreAtIvE, a competition for automated gene/protein name identification and mapping. For the gene mention task (i.e., gene/protein name identification), we used BioThesaurus, a collection of synonyms for all protein records in UniProtKB, and Metathesaurus, a collection of synonyms for medical concepts available at the Unified Medical Language System (UMLS). The machine learning task for gene mention was defined by i) transforming each word into a feature vector consisting of various types of features, and ii) training a classification system using Conditional Random Field (CRF) to classify each word to three categories: *B* word (beginning of a gene mention phrase), *I* word (inside of a gene mention phrase), and *O* word (outside of a gene mention phrase). For the gene normalization task, we assembled a dictionary consisting of synonyms for each gene record from online resources such as BioThesaurus and HUGO, conducted flexible dictionary lookup, and obtained a list of mapping pairs (Phrase, EGID), where Phrase is a term in text and EGID is one of the associated Entrez gene identifiers. We then defined a machine learning task to classify each mapping pair as Positive or Negative. Features were derived based on the mapping information related to Phrase and EGID in the corresponding document. We experimented with various machine learning algorithms available in Weka, a machine learning software package written in JAVA, and chose the one with the best performance (i.e., Bagging on Decision Tree). Our system achieved F-measures of over 85% for the gene mention task and around 78% for the gene normalization task.

Keywords: gene mention, gene normalization, machine learning, online resources, literature mining

1. Introduction

One crucial requirement for literature mining applications in the biomedical domain is the ability to accurately recognize terms that represent genes or proteins in free text (note that we use the words “term” and “name” interchangeably in the text) (Krauthammer and Nenadic, 2004; Yeh, et al., 2005). We refer to this task as biological entity name identification. Another requirement is the ability to associate these names with corresponding entries in biomedical databases (Hirschman, et al., 2005; Jenssen, et al., 2001). Such a task is called biological entity name normalization. Methods for biological entity name identification can be categorized into three ways: i) using a dictionary and a mapping method (Hanisch, et al., 2003; Hanisch, et al., 2005; Hirschman, et al., 2005; Jenssen, et al., 2001), ii) marking up terms in the text according to contextual cues or specific verbs (Fukuda, et al., 1998; Sekimizu, et al., 1998; Tanabe and Wilbur, 2002), and iii) applying machine learning algorithms on a gene/protein name annotated corpus (Hirschman, et al., 2005; Yeh, et al., 2005; Zhou, et al., 2004). As described by Hirschman et al. (Hirschman, et al., 2005), biological entity name normalization can be divided into several steps: 1) identifying gene occurrences in the text, ii) associating gene occurrences to one or more unique gene identifiers, iii) selecting the correct identifier in case of ambiguity, and iv) assembling the final gene list for each abstract. In the first BioCreAtIvE workshop, a number of teams achieved F-Measures of 80% for the biological entity name identification (i.e., Task 1A), where systems based on machine learning approaches, various features, and external knowledge sources,

combined with post-processing methods, achieved the best performance. For the biological entity name normalization task (i.e., Task 1B) evaluated using data sets obtained from model organism databases (i.e., yeast, mouse and fly), identifying gene occurrences in text can be classified into two groups: i) matching against the lexical resource (Crim, et al., 2005; Liu, et al., 2004), and ii) using the results obtained in Task 1A. After a simple table lookup with synonym lists assembled, the methods for selecting unique identifiers fell into two categories: prune the lexical resource by removing ambiguous lexical entries, or perform word sense disambiguation. Most systems employed thresholds to select final lists and one system applied a maximum entropy classifier for removing bad matches. The precision and recall rates reported for Task 1B ranged from a maximum of 92% F-Measure for yeast to 79% for mouse. We used a flexible dictionary-lookup method for Task 1B where the dictionary consists of synonyms obtained from online resources. The system achieved the best recalls for yeast and mouse but the precisions were very low. We found that using an extensive list of synonyms could improve recall while word sense disambiguation would be critical to improve the precision as also indicated by Hirschman (Hirschman, et al., 2005).

For the Second BioCreAtIVe workshop, we integrated machine learning with the dictionary-lookup methods, and submitted results for both the gene/protein name mention task and the gene/protein name normalization task. The following described our systems in detail.

2. Gene/protein name mention

2.1. System description

Our method for gene/protein name mention includes three steps. The first step is dictionary-lookup where terms in the text are looked up in a dictionary that consists of terms from two terminology sources, BioThesaurus (Liu, et al., 2006) and Metathesurus (Bodenreider, 2004). The second step is machine learning that integrates part of speech (POS) information and contextual features. The POS information was obtained using GENIA tagger (<http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/>). As the machine learning component, Conditional Random Field (CRF) implementation of Mallet (<http://mallet.cs.umass.edu>) was used. The last step is post-processing that corrects apparent errors and incorporates abbreviation information.

Dictionary lookup – We first filtered and normalized all terms in BioThesaurus and MetaThesaurus. The following terms were filtered: i) phrases whose occurrences are predominantly false positives in the training corpus (e.g., “protein”, “gene”, “IL”), ii) nonsensical phrases in BioThesaurus (e.g., “hypothetical protein” see (Liu, et al., 2006) for details), and iii) names in Metathesaurus with semantic categories irrelevant to gene/protein names (e.g., body location). Each term was then normalized by ignoring case-differences, punctuations, and lexical variations. Additionally, all Greek words were normalized to alpha and all numbers normalized to 9.

Machine learning - After the dictionary-lookup, each word W in a sentence was transformed to a feature vector, which includes W itself as well as the following features:

- 1) Dictionary lookup result of W – To mark each word with respect to dictionary lookup results, we used the common BIO notation (B – the word is the beginning of a name, I – the word is inside of a name, and O – the word is outside of a name) where B and I tags are followed by looked-up phrase semantic types. For example, B-Metathesaurus:aapp indicates the word is the beginning of a name in Metathesaurus with a semantic category as aapp (stands for Amino Acid, Peptide or Protein).
- 2) POS information of W – the POS information was obtained from the GENIA tagger.
- 3) Neighboring words of W – Neighboring words within a window size.
- 4) Token shape – this feature is similar to word formation features introduced in (Zhou, et al., 2005).
- 5) Suffix of length four – the last four letters of W, if exist.

After transforming each word into a feature vector, we then used CRF implemented in Mallet to build a classifier and classifying each word into one of the three categories (B, I, or O).

Post-processing - The post processing step corrects tagging inconsistency including the following:

- 1) Phrases containing mismatched parentheses. For example, we post-processed the following gene mention in text (denoted by the tag *gene*) “<*gene*>HMGR1 mRNA (HMGR1S mRNA</*gene*>)” to two mentions “<*gene*>HMGR1 mRNA</*gene*> (<*gene*>HMGR1S mRNA</*gene*>)”.

- 2) Phrases that differ from other tagged phrases in the same excerpt only by numbers or Greek letters. For example, if the phrase “C/EBP alpha” is tagged in an excerpt, “C/EBP beta” in the same excerpt should also be tagged, and vice versa.
- 3) Acronym/abbreviation phrases with their corresponding definitions can be detected using a procedure similar to (Schwartz and Hearst, 2003). For example, in the following excerpt “A platelet-derived growth factor receptor (PDGF-R) phosphopeptide containing Tyr-857 does not bind appreciably to the Src SH2 domain, suggesting it is not the PDGF-R binding site for Src as previously reported.”, “platelet-derived growth factor receptor” is detected as the definition for “PDGF-R”. If it is identified as a gene/protein name, then all occurrences of “PDGF-R” in the same excerpt will be tagged as gene/protein mentions.

2.2. Submission description

We submitted three runs for the gene mention task where the first run is from the base system. With the aim to improve recall, we implemented two additional procedures: the first one incorporated contextual information of the whole abstract, and the second one incorporated the results from a machine learning-based tagger provided in the LingPipe suite (<http://www.alias-i.com/lingpipe/>).

The first procedure is based on the assumption “one sense per discourse”: if a phrase is detected as a gene/protein name in an abstract, all occurrences of the phrase in that abstract are considered as gene/protein name mentions. For each excerpt, we searched for the potential source abstract in PubMed using NCBI tools with a query consisting of all the words in the excerpt. We then processed the abstract using our base system if applicable. The phrases detected as genes/proteins in the abstract were looked up in the excerpt and they, if found, were added to the first submission, and submitted as the second submission.

The third submission is to supplement the first submission with phrases that were mapped to BioThesaurus entries by dictionary-lookup and also tagged by another name recognition system, a long-distance character language model-based chunker, in the LingPipe suite [4]. This is based on the observation that some of the (true) gene/protein phrases were initially mapped to BioThesaurus entries during dictionary lookup, but they were (falsely) untagged by the base system probably due to the lack of sufficient orthographic features for machine learning.

3. Methods for gene/protein name normalization

3.1. System description

The base gene/protein name normalization system includes three modules. The first module is dictionary-lookup where the dictionary consists of terms associated with human Entrez gene records. The second module is machine learning that integrates the results of our gene/protein name mention tagger, name sources, name ambiguity, false positive rates, popularity, and token shape information. The third module is a similarity-based method to associate Entrez gene records with long phrases detected by the gene/protein name mention tagger.

Dictionary-lookup - Based on the cross-reference information in BioThesaurus and Entrez Gene releases, we obtained a dictionary consisting of synonyms for each Entrez gene record. We then performed flexible dictionary-lookup, and a list of pairs (Phrase, EGID) were obtained, where Phrase is a text string in a document mapped to a dictionary entry and EGID is the Entrez Gene identifier. If the string contained specialized patterns which usually were abbreviated forms for several entities from the same family (e.g., “HAP2, 3, 4” or “HAP2-4”, “HAP-2, -3, and -4”, or “HAP2/4”), we separated them and reassembled to several strings and tried to find mapping for each of them. For example, “HAP2/4” would become two strings “HAP2” and “HAP4”.

Machine learning - For each pair (Phrase, EGID), we extracted a list of features and defined a machine learning task. The features include:

- 1) Entity – the value is true if Phrase is detected as a gene/protein name mention by our gene mention system.
- 2) ExactMatch – the value is true if Phrase is an exact match for EGID
- 3) Ambi - the number of EGIDs associated with Phrase in the collection. This feature captures the

ambiguity of Phrase.

- 4) StrNum – the number of different phrases in the abstract corresponding to EGID. If multiple phrases are normalized to the same gene in the abstract, then it is likely the mapping is correct.
- 5) AssocDistance – the difference of the association power of Phrase to EGID to the gene record which with the maximum association power. Phrase can be associated with multiple gene records, and for each record, it can be a primary name, a synonym/alias, or a description in online resource(s). We measure the association power between Phrase and each gene record using a score which depends on the number of online resources associating Phrase with the record as well as the fields in the corresponding resource: if Phrase is the primary name or symbol for the record in a resource, the score is increased by 3; if Phrase is a synonym/alias, the score is increased by 2; otherwise, the score is increased by 1. Let MSCORE be the maximum score among all pairs of Phrase. We then consider the difference of the score associated with (Phrase, EGID) to MSCORE as a measure for disambiguation. The value of 0 indicates the strongest association of Phrase to the gene record comparing to others. The higher the value is, the less chance for (Phrase, EGID) to be true.
- 6) Primary, Description, Synonym – if one of the online resources considers Phrase as a primary name or symbol for EGID, Primary is true; otherwise, if Phrase comes from non-name fields of all online resources, Description is true; otherwise, Synonym is true.
- 7) FPRate - The false positive rates on the noisy training data. We conducted the dictionary-lookup on the noisy training data and computed the false positive rates associated with the pair.
- 8) EGIDFreq - The occurrences of all phrases associated with EGID in the document.
- 9) PhrFreq - The frequency of Phrase in the document. This feature is intended to see how frequent Phrase appeared in the document. The more frequent it is, the more likely it is a name.
- 10) PhrGFreq - The frequency information for Phrase obtained from the top 100,000 word list of MedPost. This feature is intended to capture the occurrences of the phrases in the whole MEDLINE collection.
- 11) GreekNum – if Phrase contains numbers (i.e., “1”, or “I”) or Greek letters (i.e., “alpha”, “beta”)
- 12) MixCase – if Phrase contains both upper- and lower cases letters.
- 13) pLeft (pRight) – the value will be true if the immediate left (or right) character is non-space punctuation.
- 14) sLeft (sRight) – the value will be true if the immediate left (or right) character is a space.

Since each pair was transformed to a fixed set of features, almost all standard machine learning algorithms can be used.

Similarity-based mapping – Names with multiple words in a dictionary may appear in the text with some of the words missing, or in different word orders or forms. For example, in the training set, gene GLRA1 (EGID is 2741) is one of the genes mentioned in the abstract (PMID: 8651283). The phrase in text is “human glycine receptor (GlyR) alpha 1 subunit gene” which cannot be mapped to any of the synonyms we collected for gene GLRA1. However, all of the following words, “glycine”, “receptor”, “alpha”, “1”, and “subunit”, appear in names for gene GLRA1 in our dictionary. We incorporated a similarity-based method for normalizing names detected by our gene/protein mention tagger. We counted the number of words overlapped between phrases detected as entity names in text and names in the dictionary. If over 90% of the words in a name from the dictionary can be found in the names detected by the gene/protein name tagger, we consider the names in the text can be normalized to associated record(s) of the name. In the above example, all words in name “glycine receptor, alpha 1” (which is the Entrez gene description) can be found in the phrase, the similarity-based method will normalize the phrase to gene GLRA1.

3.2. Submission description

We experimented with various machine learning algorithms available in the software package Weka. Based on the performance of the ten-fold cross validation on the training data, we selected “bagging on decision tree” as the final machine learning algorithm since it achieved the best performance. We submitted three runs for the normalization task. Two out of the three runs (the second and third runs) were obtained using different versions of the dictionary: i) the strict version where we filtered out names that are frequent common English words and also names that are only associated with false positives in the noisy training data, and ii) the raw version which contained all names assembled. The first run was the combination of the second and third runs. For all submissions, associations identified using the similarity-based mapping method were included.

Table 1: Gene mention (GM) and gene normalization (GN) results.

	Precision (Quartile)	Recall (Quartile)	F-Measure (Quartile)
GM-Run1	0.857 (2)	0.848 (2)	0.853 (2)
GM-Run2	0.834 (3)	0.880 (1)	0.856 (2)
GM-Run3	0.827 (3)	0.893 (1)	0.859 (2)
GN-Run1	0.743	0.824	0.781 (1)
GN-Run2	0.764	0.792	0.778 (1)
GN-Run3	0.790	0.769	0.779 (1)

Table 2: Error analysis results for gene mention (GM).

	GM-Run1		GM-Run2		GM-Run3	
	#FP (%)	#FN(%)	#FP(%)	#FN(%)	#FP(%)	#FN(%)
Over-extended Boundary	180 (20.2)	~180 (18.7)	181 (16.3)	~181 (23.9)	181(15.3)	~181 (26.9)
Under-extended Boundary	129 (14.4)	~129 (13.4)	150 (13.5)	~150(19.8)	193 (16.3)	~193 (28.7)
Ambiguous Short Forms	307 (34.4)	367 (38.2)	465 (42.0)	265 (35.0)	494 (41.8)	224 (33.3)
Others (e.g., generic)	277 (31.0)	285 (29.7)	312 (28.2)	162 (21.4)	313 (26.5)	74 (11.0)
Total	893	961	1108	758	1182	672

4. Results and Discussions

Table 1 summarizes our results. The system achieved F-measures of the second quartile among 21 teams for the three submissions of the gene mention task and F-measures of the first quartile among 20 teams for the three submissions of the gene normalization task. For the second and third submissions of the gene mention task, we received the first quartile recall measures.

Table 2 summarized error analysis results. Note that we considered a phrase as a short form (i.e., symbol/abbreviation/acronym) if it contains at most four alphabetic letters. The number inside parentheses indicates the percentage. For example, 180 (20.2%) of the false positives in the first submission were over-extended boundary errors, 129 (14.4%) were under-extended boundary errors, 307 (34.4%) consisted of at most four alphabetic letters (i.e., short forms).

From Table 2, we found two main types of errors: i) boundary detection errors, and ii) ambiguous short forms. Some of the boundary detection errors can be considered partially correct. The following shows some examples:

1. **Left boundary over-extended** – false positive “transcription factor PU.1” vs. false negative “PU.1”
 2. **Right boundary over-extended** -false positive “v-rasHa retrovirus” vs. false negative “v-rasHa”
 3. **Left boundary under-extended** - false negative “GTP-binding Ypt1 protein” vs. false positive “Ypt1 protein”
 4. **Right boundary under-extended** – false negative “ribulose-1,5-bisphosphate carboxylase/oxygenase (Rbu-P2 carboxylase) activase” vs. false positive “ribulose-1,5-bisphosphate carboxylase/oxygenase”.

The ambiguity of Symbols/Abbreviations/Acronyms is another major cause of the detection errors for both false positives and negatives. For example, in the following excerpt, "This suggests that the duration of varicocele per se could affect DHT seminal plasma levels." [PMID 6638539], "DHT" is a symbol of a steroid hormone (i.e., Dihydrotestosterone) but was detected as a gene/protein phrase by our system. A semantic type classification system is needed in order to resolve such ambiguity.

The remaining false positives include non-specific mentions of entities such as “mouse genomic sequence” and “unusual tRNA-like sequence.” But it is not always consistent. For example, “94-K transgenes” is a false positive, while “Lin-59 transgenes” is a true positive. During the error analysis, we also found that

alternative boundaries in the gold standard are not always consistent. For example, both phrases “endogenous PKR” and “PKR” are considered correct while “endogenous alpha-ENaC gene” was considered as a false positive and “alpha-ENaC gene” was considered as a false negative during the evaluation.

For the gene normalization task, we found that the F-measures obtained from the strict and raw versions of the dictionary were almost the same. The finding indicates that a rich set of synonyms can be used as is for the gene normalization task when an appropriate machine learning task is defined. We used an extensive list of features but currently we are not clear on the extent to which each individual feature contributes, although they all seem to have contributions. In the future, we plan to conduct error analysis and study the contribution of each individual feature when the gold standard list becomes available.

5. Conclusion

Utilizing machine learning and online resources, we obtained encouraging results for both the gene mention and gene normalization tasks. However, the system is based on annotated corpora which are expensive to obtain. In the future, we plan to use online resources to automatically obtain annotated corpora to build machine learning systems for the gene mention and/or normalization tasks.

Acknowledgement The project was supported by IIS-0639062 from the National Science Foundation.

Reference

- Bodenreider, O. (2004) The Unified Medical Language System (UMLS): integrating biomedical terminology, *Nucleic Acids Res*, **32**, D267-270.
- Crim, J., McDonald, R. and Pereira, F. (2005) Automatically annotating documents with normalized gene lists, *BMC Bioinformatics*, **6 Suppl 1**, S13.
- Fukuda, K., Tamura, A., Tsunoda, T. and Takagi, T. (1998) Toward information extraction: identifying protein names from biological papers, *Pac Symp Biocomput*, 707-718.
- Hanisch, D., Fluck, J., Mevissen, H.T. and Zimmer, R. (2003) Playing biology's name game: identifying protein names in scientific text, *Pac Symp Biocomput*, 403-414.
- Hanisch, D., Fundel, K., Mevissen, H.T., Zimmer, R. and Fluck, J. (2005) ProMiner: rule-based protein and gene entity recognition, *BMC Bioinformatics*, **6 Suppl 1**, S14.
- Hirschman, L., Colosimo, M., Morgan, A. and Yeh, A. (2005) Overview of BioCreAtIVe task 1B: normalized gene lists, *BMC Bioinformatics*, **6 Suppl 1**, S11.
- Hirschman, L., Yeh, A., Blaschke, C. and Valencia, A. (2005) Overview of BioCreAtIVe: critical assessment of information extraction for biology, *BMC Bioinformatics*, **6 Suppl 1**, S1.
- Jenssen, T.K., Laegreid, A., Komorowski, J. and Hovig, E. (2001) A literature network of human genes for high-throughput analysis of gene expression, *Nat Genet*, **28**, 21-28.
- Krauthammer, M. and Nenadic, G. (2004) Term identification in the biomedical literature, *J Biomed Inform*, **37**, 512-526.
- Liu, H., Hu, Z., Torii, M., Wu, C. and Friedman, C. (2006) Qualitative Assessment of Dictionary-based Biological Named Entity Tagging, *JAMIA*, 13, 497-507.
- Liu, H., Hu, Z.Z., Zhang, J. and Wu, C. (2006) BioThesaurus: a web-based thesaurus of protein and gene names, *Bioinformatics*, **22**, 103-105.
- Liu, H., Wu, C. and Friedman, C. (2004) BioTagger: a biological entity tagging system. *BioCreAtIVe Workshop*. Spain.
- Schwartz, A.S. and Hearst, M.A. (2003) A simple algorithm for identifying abbreviation definitions in biomedical text, *Pac Symp Biocomput*, 451-462.
- Sekimizu, T., Park, H.S. and Tsujii, J. (1998) Identifying the Interaction between Genes and Gene Products Based on Frequently Seen Verbs in Medline Abstracts, *Genome Inform Ser Workshop Genome Inform*, **9**, 62-71.
- Tanabe, L. and Wilbur, W.J. (2002) Tagging gene and protein names in biomedical text, *Bioinformatics*, **18**, 1124-1132.
- Yeh, A., Morgan, A., Colosimo, M. and Hirschman, L. (2005) BioCreAtIVe task 1A: gene mention finding evaluation, *BMC Bioinformatics*, **6 Suppl 1**, S2.
- Zhou, G., Shen, D., Zhang, J., Su, J. and Tan, S. (2005) Recognition of protein/gene names from text using an ensemble of classifiers, *BMC Bioinformatics*, **6 Suppl 1**, S7.
- Zhou, G., Zhang, J., Su, J., Shen, D. and Tan, C. (2004) Recognizing names in biomedical texts: a machine learning approach, *Bioinformatics*, **20**, 1178-1190.



Me and my friends: gene mention normalization with background knowledge

Jörg Hakenberg¹

hakenbergj@biotec.tu-dresden.de

Loic Royer¹

loic.royer@biotec.tu-dresden.de

Conrad Plake^{2,1}

cplake@transinsight.com

Hendrik Strobelt¹

hendriks@biotec.tu-dresden.de

Michael Schroeder¹

ms@biotec.tu-dresden.de

¹ Biotechnological Centre, Technische Universität Dresden, Tatzberg 47–51, 01307 Dresden, Germany

² Transinsight GmbH 47–51, 01307 Dresden, Germany

“Tell me who your friends are, and I will tell you who you are” – this proverb best illustrates our approach to the normalization of gene names. In this approach, we rely on background knowledge that describes various aspects of a gene: it is localized on a chromosomal band, it belongs to an operon structure, it is a member of a gene family, its products take part in biological processes, they fulfil molecular functions, they occur at dedicated cellular locations, mutations of the gene ultimately cause diseases, its proteins contain domains and form secondary, tertiary and quaternary structures. Whenever a gene (or one of its products) is discussed, some of these aspects –the gene’s friends, that is, semantically related information– will be mentioned as well. The paradigm we follow with this approach demands not only the presence of a gene’s name, but also of some of its friends.

We see every set of information available (see Methods section) for each gene as this gene’s description, or the context this gene typically “lives” in. Whenever we encounter ambiguities regarding proper identification of a gene, we assess each potential candidate gene by comparing its typical context against the predicted one (in this case, a PubMed abstract.) The descriptions for genes originate from various curated resources: EntrezGene provides organisms, summaries, chromosomal loci, Gene Ontology (GO) terms, and encoded proteins; UniProt provides functional descriptions, protein domains, interaction partners, keywords, and GO terms; more GO annotations are provided by GOA.

Consider the example of the oncogene p54 (reflected in Figure 1.) Having resolved the issue of potential organisms, there are still human genes from EntrezGene that share the same name. Indeed, they refer to completely different genes with disjoint annotations. Based on the name alone, this problem could not be solved. Only comparing each of the gene’s contexts to the text reveals that one of the potential candidates is a RNA helicase, and the text indeed mentions “RNA helicase.” The text also mentions the exact chromosomal location of the correct gene.

A gene encoding a putative **human RNA helicase, p54**, has been cloned and mapped to the band **q23.3 of chromosome 11**. The predicted amino acid sequence shares a striking homology (75% identical) with the female germline-specific **RNA helicase ME31B** gene of **Drosophila**. Unlike ME31B, however, the new gene expresses an abundant transcript in a large number of adult tissues and its 5' non-coding region was found split in a **t(11;14)(q23.3;q32.3)** cell line from a diffuse large B-cell lymphoma.

EntrezGene ID: 1656
P54; RCK; HLR2
Species: **H. sapiens**
Chromosome: 11q23.3
GO: **RNA Helicase**

EntrezGene ID: 2289
P54; FKBP51; PPIase
Species: **H. sapiens**
Chromosome: 6p21.3-2
GO: isomerase activity

EntrezGene ID: 4841
P54; NMT55; NRB54
Species: **H. sapiens**
Chromosome: Xq13.1
GO: RNA splicing

EntrezGene ID: 42828
S4; dRpt2; **p54**; p56
Species: **D. melanogaster**
Chromosome: 3R;95C13
GO: proteolysis

Figure 1: The given abstract as a whole points out one out of four genes: synonym, species, chromosomal location, biological process all fit best to the leftmost gene context (PubMed ID: 1579499).

Methods

The system we propose for identification of gene names in texts consists of four major components. The basic step provides an initial recognition of candidate terms, which also assigns all potential EntrezGene IDs to each candidate. From there on, the next components deal with refining these candidate hits: removal of false positives and disambiguation of polysemous names. The second component finds text parts that never contain a gene name but might account for errors of the recognition step. The third component filters false positives by looking at term frequencies, and reduces the candidate IDs by comparing new to known texts (from the “noisy” training data.) The final component disambiguates remaining terms and identifiers using each gene’s typical context. On the BioCreative2 GN test set, our system achieves an F1-measure of 81% (highest recall: 87.5%, highest precision: 79%).) The highest recall we measured on the training data set was 92.7% (at <40% precision); this was achieved when not using the disambiguation.

Description of the submitted run	Precision	Recall	F1 (in %)	TP	FP	FN
NER with extended masterlist, FP+FN filter, disambiguation	78.9	83.3	81.0	654	175	131
NER with extended masterlist, FP filter, no disambiguation	49.6	87.5	63.3	687	699	98
NER with unextended masterlist, FP filter, disambiguation	70.7	72.5	71.6	569	236	216

Named entity recognition

For the initial recognition of potential gene names and their EntrezGene identifiers, we extended the provided masterlist with additional synonyms found on the EntrezGene website, plus synonyms for the gene products. We then sorted each synonym into one of four categories:

- database identifiers (“KIAA0958”, “HGNC:17875”),
- abbreviations (“CD95L”, “Lin7c”),
- single- or multi-word terms (“tumor necrosis factor alpha”, “RAD51-interacting protein”), and
- spurious synonyms (“AA”, “ORF has no N-terminal ‘Met’, it may be non-functional”).

We ignored spurious synonyms in the remainder, as they never occur in text, but only in database fields. For each synonym class we applied specialized search strategies.

- Database identifiers were extracted using regular expressions, yielding immediate identification: “KIAA0958” could appear as “Kiaa0958”, yet it was unique and pointed to a single ID.
- Abbreviations got segmented around optical gaps: white spaces, punctuation, transitions between digits, lower case or upper case letters. We generated variations for each segment and re-combined them. Variations affected case changes, transformations between Latin/Arabic/Greek/English, and structural changes (“CD95 receptor”, “receptor of CD95.”) Starting with the known synonym “IFN-gamma”, the mentioning “Ifng” has to be recognized.
- Multi-word terms were tokenized and each token was evaluated for potential variations, comparable to the abbreviation class. This added possible spelling variants of each synonym. Some tokens were optional and not essential for recognition (“protein” at the end of a name), because they often are omitted in text.

Each synonym could correspond to several different genes and thus different identifiers. To remove obvious false positives, we used a filtering algorithm based on contextual rules. Each rule was a triplet consisting of three regular expressions, the first matching the context immediately before a potential gene name, the second matching the name itself, and the third matching the context right after the name. For example, an initial candidate name immediately followed by “cells” most likely referred to a cell line, and only implicitly to a gene/protein. “Mouse” before a name hinted to a mouse gene, but if the name was then followed by “homolog”, this rule did not apply immediately. We created these rules manually driven by examples from the training data.

The last step of this initial recognition merged consecutive candidate names (that shared one identifier) into one contiguous candidate. Such occurrences were most likely to refer to one and the same gene. Such tuples appeared, for instance, when abbreviations were introduced and a long form was followed by its abbreviation in brackets. We kept only such EntrezGene IDs that were assigned to all consecutive candidates; we kept the IDs of the long form when there were no IDs common to

all, dropping all other IDs. In addition, we expanded ranges (such as in “seven novel forkhead genes, freac-1 to freac-7”) to the full list of all names included therein.

Disguising false positive sites

The second component of our system marked obvious irrelevant parts that often accounted for false positives. It removed the following types of phrases prior to NER: units like “497 amino acids”; cell types and descriptions (“CD34+”); DNA/RNA (“ACGGT”, “cDNA”); chromosomal locations (“chromosome 20 on band p13”, “21q22.1”); and abbreviations not related to genes/proteins (“Human granulocytic ehrlichiosis (HGE”)). This avoided some errors introduced by the first component, for instance, the detection of “p13” in the chromosomal location example. As another filter, we removed unspecific references (to protein families etc.) from the predicted candidate names. We noticed that in most cases, (even multi-word) names that consisted entirely of lower case letters could also be removed.

Identification of candidates

After the initial recognition of gene names, we proceeded to identify each name. We passed the annotated texts through several filters to reduce the number of possible IDs for each gene name and to find the correct masterlist entry: We first searched for exact matches of candidate names in the masterlist. In case only one entry was found, we took this entry directly as the annotation. For ambiguous cases (multiple entries for the name), we compiled a set of representative texts for each entry from the noisy data and EntrezGene Summary. From 8243 ambiguous entries, 2954 had abstracts in the noisy training data (see GN task description), 3906 had an EntrezGene Summary, and 2074 had both. Every set of texts was transformed into a set of feature vectors with tf-idf feature weights. We then searched for the 100 abstract most similar to the current abstract (cosine-based distance.) From the set of IDs resulting from this comparison (each of the 100 representatives had one or more genes assigned), we selected the subset of IDs that had synonyms matching the candidate gene name. For matching, we used an approximative, character-based alignment. All IDs from this subset were taken into further consideration. To all remaining gene names we assigned a tf-idf score based on the current abstract and the overall text corpus. If a candidate name achieved a low tf-idf score, we dropped it as a likely false positive annotation. This step thus dealt with two types of errors introduced by the named entity recognition: it removed false annotations and it found genes initially missed.

Disambiguation by candidate ranking

The fourth component disambiguated each polysemous name. We compared background knowledge available for each gene (gene context) with the current text and picked the gene which context best fitted the current text. We collected external knowledge from EntrezGene, UniProt, and GOA for each of the 30,000 genes (EntrezGene: summary, GO terms; UniProt: diseases, keywords, functions, GO terms; GOA: GO terms.) For EntrezGene and UniProt, we calculated the overlap of the text at hand with each annotation based on tokens. For calculating the similarity based on GO terms, we used GoPubMed to find GO terms in the current text [1]. For each potential tuple taken from the two sets (text & gene annotation), we computed a distance of the terms in the ontology tree (comparable to [2]). These distances yielded a similarity measure for two terms, even if they did not belong to the same sub-branch or were immediate parents/children of each other. The distance took into account the shortest path via the lowest common ancestors, as well as the depth of this LCA in the overall hierarchy. All five comparisons yielded likelihoods stating the similarity of the current text with the knowledge available on each gene. We combined the likelihoods into confidence measures, and picked the EntrezGene ID with the highest probability, if this was above a certain threshold.

References

- [1] Doms, A. and Schroeder, M., GoPubMed: exploring PubMed with the Gene Ontology. *Nucleic Acids Res.*, 33:W783–786, 2005.

- [2] Schlicker, A., Domingues, F.S., Rahnenführer, J., and Lengauer, T., A new measure for functional similarity of gene products based on Gene Ontology, *BMC Bioinformatics*, 7:302, 2006.



Context-Aware Mapping of Gene Names using Trigrams

ThaiBinh Luong^{1,2}
thaibinh.luong@yale.edu

Nam Tran¹
nam.tran@yale.edu

Michael Krauthammer^{1,2}
michael.krauthammer@yale.edu

¹ Department of Pathology, Yale University, New Haven, CT, USA

² Program for Computational Biology and Bioinformatics, Yale University, New Haven, CT, USA

Abstract

We present a method for the mapping of gene names to Entrez Gene identifiers. We first resolve lexical variation by transforming domain terms into their unique trigrams, and use this representation for a preliminary term mapping. We then perform fine-mapping via contextual analysis of the abstract that contains the domain term. We have formalized our method as a sequence of matrix manipulations, allowing for a fast and coherent implementation of the algorithm. We pair our method with existing approaches for entity recognition, and achieve an F-score of 0.761 in the BioCreative 2 Gene Normalization Task.

Keywords:

1 Introduction

Our paper addresses the Gene Normalization Task of the BioCreative 2 Challenge. We approach this task as a term identification problem, which can be subdivided into three modular stages: term recognition, term classification, and term mapping [1]. Our method presented here focuses on the third step, the mapping of biomedical terms to some controlled vocabulary, which we think is most relevant with respect to the Gene Normalization Task. The advantage of our approach is that our mapping strategy is independent from the underlying term recognition and classification process, and can therefore be paired with a multitude of previously published methods for recognizing and classifying terms.

We believe that two fundamental processes are at play when mapping biomedical terms: A first *approximate* mapping of a term to known biomedical concepts, and a subsequent fine-mapping using contextual analysis. The first step analyzes lexical term variation, and results in a prioritized list of possible biomedical concepts. It is solely based on the local features (aka the letters/words) of the unmapped biomedical term. The second step is a contextual analysis of the term mentioning. Only the latter enables the definite placement of the term with respect to a unique biomedical concept. We believe that this approach may be similar to the way we humans approach the term mapping problem. After encountering a novel gene name, which looks similar (but not identical) to known gene names, we can infer the correct gene by comparing the context (such as a scientific abstract) with the previously encountered literature. The contextual analysis may result in the identification of similar, already known abstracts, that discuss known genes. If the novel gene name is similar to the names of those known genes, we can easily make the final term assignment.

There are several noteworthy features of our approach: First, we are clearly separating the local and contextual mapping, enabling the experimental examination of both processes individually. Second, our local analysis is fast and efficient, avoiding the traditional string matching techniques. Similarly, we perform a fast contextual analysis with respect to thousands of previously published abstracts.

2 Methods and Results

As discussed above, we approach the task as a term mapping problem. The idea is to use existing programs for entity recognition, and then use the methods described below to map recognized and classified strings to external gene identifiers (in our case: Entrez GeneIDs). For entity recognition, we use Abner [2] (both Biocreative and NLPBA settings) and LingPipe¹ (GeneTag model), two programs with excellent recall and precision. We process PubMed abstracts three times, for each program and setting. Each run gives rise to a separate list of recognized entities, which are then separately mapped to Entrez GeneIDs. A majority vote is then cast to determine the list of abstract-specific GeneIDs.

We use a combination of two methods to map recognized entities to their appropriate gene identifiers: the *Trigram Method*, and the *Network Method*. Both methods require preprocessing, using resources from Entrez Gene, to construct a set of method-specific matrices.

2.1 Trigram Method

The first method, as mentioned earlier, is designed to quickly retrieve a list of possible gene identifiers, which are good mapping candidates for each entity recognized by Abner/LingPipe. The method should be fast, but does not need to resolve uncertainties, such as homonymy. In short, our method utilizes an approximate representation of a gene names, by transforming a name into the set of its unique trigrams. The similarity between 2 gene names is the number of their common trigrams (i.e. the intersection of their sets of trigrams). This approach allows for the fast mapping of a gene name to a dictionary of gene names, such as the Entrez Gene resource, with its associated gene identifiers.

To accomplish this, we first need a preprocessing step, in which all the unique gene names/synonyms (“gene strings”) from the Entrez Gene resource are identified, and split into a set trigrams, a succession of three alphanumeric characters. For example, the gene string “lypla1” (the official symbol of GeneID 10434) would be split into 4 trigrams: “lyp”, “ypl”, “pla”, and “la1”.

Let m be the number of all the possible trigrams (that occur across all strings in the Entrez Gene resource), then a string s is represented by an m -vector v of 0 and 1, such that $v_i = (i\text{th trigram } \in s)$ for all $1 \leq i \leq m$.

The similarity between two strings u and v is defined as the dot product $u \cdot v$.

Let n be the number of all the unique Entrez Gene strings. Let A_S be the $n \times m$ matrix whose rows are the vector transposes of the strings’ representations. We can then easily determine the similarities of a query string u (i.e. the trigram representation of the string recognized by Abner/LingPipe) to all the Entrez Gene strings by computing the product

$$r_S = A_S u$$

The results vector r_S is of dimension n , the number of unique gene strings. The similarity scores need to be normalized, in order to penalize improper string matches. For example, suppose our query string is “abl”. Gene strings that contain words such as “transposable”, “disable”, or “variable” will receive the same similarity scores as a simple gene string “abl”. For this reason, we take into consideration how well a query string is contained within an Entrez Gene string, ie whether the number of trigrams in the query string matches the number of trigrams in the gene string. Vice versa, we also calculate how well an Entrez Gene string is contained within the query string. We thus weight the results vector r_S accordingly, assigning the highest weights to gene strings that match the query string exactly (are perfectly contained within each other). We denote the normalized results vector r_{S_n} . The latter vector contains similarity scores for each gene string. However, we are interested in finding the maximum similarity score on the gene level, i.e. looking at each synonym of a gene (a set of gene strings) and selecting the synonym (gene string) with the highest score. This is done by probing results vector r_{S_n} in a gene-by-gene fashion. To accomplish this, we construct an $n \times l$ matrix, A_{GS} , where l is the number of unique GeneIDs, and n is the number of unique Entrez Gene strings as described above.

A value of “1” in $A_{GS(i,j)}$ implies that GeneID j is associated with gene string i . We then update A_{GS} by r_{S_n} .

$$A_{GSu} = \text{diag}(r_{S_n}) A_{GS}$$

From A_{GSu} , we construct a vector g_S , which is of size l , the number of unique (human) GeneIDs.

¹ <http://www.alias-i.com/lingpipe>

$$g_S = [\|A_{GSu}^{(1)}\|_\infty, \|A_{GSu}^{(2)}\|_\infty, \dots, \|A_{GSu}^{(l)}\|_\infty]$$

Here, $A_{GSu}^{(i)}$ is the i th column vector of A_{GSu} and $\|\cdot\|_\infty$ is the maximum norm. Thus $g_{S(j)}$ represents the highest scoring gene string per GeneID j .

2.2 Network Method

The first method calculates g_S , a vector of size l , the number of human genes, with $g_{S(i)}$ representing the trigram-similarity score of gene i (with respect to a recognized entity E). It is possible that several genes have the same similarity score, and we need another method for pinpointing the correct gene identifiers. To accomplish this, the Network Method examines the words (context) of the abstract, where the entity has been recognized. The idea is as follows: Assume that the Trigram Method determines that a recognized entity E may be linked to two different gene identifiers (gene A and B) with equal similarity scores. The network method compares the abstract a , where the entity has been recognized, to a collection of abstracts where gene A and B have been positively identified. If the content of abstract a is closer to the set of abstracts linked to gene A , we label entity E with gene identifier A . We devised a method to rapidly perform the above procedure across all human genes. As in the Trigram Method, there is a need to preprocess external resources to create method-specific matrices. We use the Entrez gene2pubmed resource to identify p abstracts that are positively linked to human genes (often, several abstracts are linked to a single human GeneID). We preprocess those abstracts to extract a list of unique and stemmed words, and weigh those words according to a normalized TF*IDF measure. We then construct a $p \times q$ matrix A_N , where p is the number of abstracts and q is the number of unique stemmed words that appear across all p abstracts. Furthermore, we construct a $p \times l$ matrix A_{GN} associating abstracts with their GeneIDs (similar to the matrix A_{GS} in the Trigram Method above). We follow a similar procedure as outlined in the Trigram Method above. Given an input abstract containing the recognized entity E , we transform the abstract into a q -vector u and calculate

$$r_N = A_N u$$

r_N is of size p , the number of abstracts, and contains the resulting similarity scores of the input abstract a to the abstracts in A_N . We then can easily² group the abstracts that are mapped to the same GeneID by calculating

$$g_N = A_{GN} r_N$$

Vector g_N is of size l , the number of unique (human) GeneIDs, and contains the similarity scores of the abstract a to each GeneID³.

2.3 Combining the Methods

The vectors of trigram scores and network scores for each Entrez Gene, g_S and g_N , are now combined to assign the final GeneID for each recognized entity E . We first look at g_S , and read the set of those GeneIDs with a perfect score of 1. If the set consists of a single GeneID, we assign that ID to the entity E . If the set is >1 , we sort the set by the network score g_N , and assign the highest ranked GeneID to the entity E . By default, we do not assign a GeneID if there is no entry in g_S with a perfect score of 1 (this measure aims at eliminating incorrectly recognized entities).

2.4 Results

We evaluated our two methods on the Biocreative 2 GN testing set, which consisted of 262 abstracts discussing human genes. The task was to identify all the gene identifiers of those genes. We submitted a single run of our program to the BioCreative Challenge. Our combination of the Trigram and Network methods yielded a recall of 0.740, a precision of 0.784, and an f-score of 0.761. Subsequent analysis of the Trigram method on the same set produced results, which were slightly lower than the Trigram/Network method, as expected. The recall and precision were 0.684 and 0.707, respectively, and the f-score was 0.695.

² Not shown is a normalization step, where we normalize g_N with respect to the number of abstracts that link to a particular GeneID.

³ The BioCreAtivE 2 GN training and testing sets contained abstracts that were part of Entrez's gene2pubmed file. We checked whether a training and testing abstract a was part of gene2pubmed, and removed the mapping information of abstract a from A_{GN} (the identification of the correct gene identifiers for abstract a would otherwise be trivial).

We also analyzed our method's ability for gene name mapping in presence of a perfectly marked-up corpus (ie perfect entity recognition), where we assign GeneIDs to all entities. Our preliminary data suggest that we can achieve an accuracy of 0.78 for mapping to the correct GeneID (unpublished data).

The main reason for mis-mapping stems from the issue of “containment”. Our computation of trigram scores favors genes that more closely contain the entity *and* do not contain extra trigrams. Another source of incorrect mapping can be attributed to the lack of a close variation in the gene_info file (our “dictionary”). The last major category of incorrect mapping are those entities in which we cannot correctly disambiguate between two genes that have the same trigram score, but very close network scores. In many of the cases, the group of lexically equivalent genes belong to the same family of genes.

3 Discussion

We describe a coherent, matrix-based method for approximate and contextual term mapping in the biomedical domain. We believe that our approach is unique in that it provides a coherent framework in solving both the problem of lexical variation and term ambiguity of gene names.

A trigram-representation of phrases has been previously described as being useful in finding synonyms in the biomedical domain [3]. Here, we show that the use of trigrams is similarly effective for mapping of gene names. Also, there exist earlier studies that discuss the inclusion of contextual information for term mapping (see for example [4, 5]). We think that our method adds an elegant solution to this problem, by providing a fast vector-space method for resolving gene name ambiguity in a large biomedical dictionary (Entrez Gene), without the need for machine learning. There is ample room for expansion of our method. We are investigating different ways of combining the results vectors g_S and g_N of the Trigram and Network method, respectively. We also need to address the problem of gene names that consist of fewer than 3 characters. An obvious solution is the use of a bigram representation.

References

- [1] Krauthammer, M. and G. Nenadic, *Term identification in the biomedical literature*. J Biomed Inform, 2004. 37(6): p. 512-26.
- [2] Settles, B., *ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text*. Bioinformatics, 2005. 21(14): p. 3191-2.
- [3] Aronson, A.R., et al., *The NLM Indexing Initiative*. Proc AMIA Symp, 2000: p. 17-21.
- [4] Liu, H., S.B. Johnson, and C. Friedman, *Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the UMLS*. J Am Med Inform Assoc, 2002. 9(6): p. 621-36.
- [5] Schuemie, M.J., J.A. Kors, and B. Mons, *Word sense disambiguation in the biomedical domain: an overview*. J Comput Biol, 2005. 12(5): p. 554-65.



ProMiner: Recognition of Human Gene and Protein Names using regularly updated Dictionaries

Juliane Fluck Heinz Theodor Mevissen Holger Dach
juliane.fluck@scai.fhg.de *theo.mevissen@scai.fhg.de* *Holger.dach@scai.fhg.de*

Marius Oster Martin Hofmann-Apitius
marius.oster@scai.fhg.de *martin.hofmann-apitius@scai.fraunhofer.de*

Fraunhofer Institute for Algorithms and Scientific Computing (SCAI)
Department of Bioinformatics
Schloss Birlinghoven, Sankt Augustin, Germany

Abstract

For the recognition of gene and protein names and their normalization to gene and protein centered databases (Entrez Gene and UniProt) regularly updated dictionaries generated from these sources are used by the ProMiner system to search gene and protein names in scientific publications. A multistep curation process and inclusion of different biomedical dictionaries in the curation process leads to an increase of precision and recall. The recognition of names containing special parenthesis expressions augments the recall further. Human gene and protein names in the test corpus provided in BioCreAtIVe II could be recognized with the adapted ProMiner system and a regularly updated dictionary with a final F-measure of 80 %.

Keywords: named entity recognition, text-mining, gene normalization

1 Introduction

The ProMiner system was developed for the automatic generation of gene and protein name dictionaries and their recognition in scientific texts. The performance of the approach taken with ProMiner was already demonstrated in BioCreAtIVe I where an F-score of 0.8 could be reached for fly and mouse and an F-score of 0.9 for the yeast organism [3]. For BioCreAtIVe II two different training corpora, an automatic generated noisy training set (5000 Medline abstracts) and a manual annotated corpus (282 abstracts) including Entrez Gene identifier of the occurring human genes were provided. The performance of the entity recognition procedure was estimated on an independent set of 262 abstracts. Annotations for these sets were not available during the competition. On basis of a gold standard provided by human experts, submitted results were assessed by the organizers.

The recognition of human gene and protein names with ProMiner has already been used in different application scenarios like the generation of disease centric databases, e.g. the Auto Immune Data Base (AIDB) [5] or an intracranial aneurysm knowledge base in the European project @neurIST¹. The ProMiner system includes an updating and dictionary curation process to generate gene and protein name dictionaries from the databases Entrez Gene [7] and UniProt [1]. Here, we describe the standard updating process for the human dictionary and adaptations made to the BioCreAtIVe sets. Furthermore, in the recognition module an extension for the recognition of names containing special parenthesis expressions is integrated.

2 The ProMiner software for recognition of gene and protein names

The ProMiner system has already been described in detail in ([2,3]). In this paper we give a short overview (cf. figure 1) over the sources used for the generation of the dictionaries, the different ProMiner modules, and the adaptations made for the BioCreAtIVe II gene normalization assessment.

The human dictionary is extracted from the gene description fields of human Entrez Gene entries and the protein description fields of human UniProt entries. All entries that are transitively mapped to each other in the International Protein Index (IPI) [6] are merged to one dictionary entry. For the BioCreAtIVe assessment

¹ <http://www.aneurist.org/>

we separate all entities containing more than one Entrez Gene entry. The dictionary used for the BioCreAtIvE assessment is based on an extraction of all files from release date 1st August 2006.

In the automatic dictionary curation, several functionalities such as acronym expansion, addition of spelling variants or filtering synonyms on the basis of regular expressions are covered. Its tasks are to add certain terms like long-forms of acronyms or spelling variant like IL1 (in addition to IL 1) to the dictionary in order to gain recall or to detect unspecific synonyms to either prune them from the dictionary (i.e. 35 kDa protein) or mark them for later processing (i.e. ambiguous synonyms). In the human dictionary, one-word synonyms are expanded with a leading „h“ (e.g. hSMRP). The new name is included (in addition to the original one) only if it is unique in the dictionary.

Additionally, a manually curated list generated through inspection of various training corpora in different former and ongoing projects (independent from BioCreAtIvE e.g. in the context of [5]) is used for curation of the human dictionary. Furthermore, we extract from the Open Biomedical Ontology (OBO) site different ontologies² for disease, tissue, organism and protein family names (BioMed terminology). In order to prune such unspecific gene and protein names all human dictionary synonyms matching in a ProMiner search to names from the BioMed terminology dictionary are removed. For BioCreAtIvE II false positive hit lists generated by ProMiner runs on the training set and the noisy training set are inspected by a curator and added to the curation list.

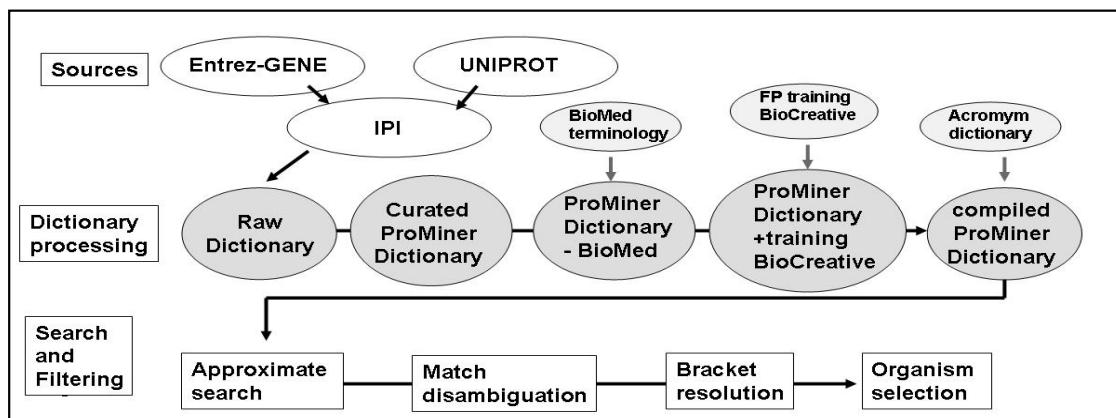


Figure 1: The ProMiner system used in BioCreAtIvE gene normalization

Finally an algorithm similar to [4] is used to extract acronyms and their long forms from all MEDLINE abstracts, generating an *acronym dictionary*. A *gene search specific acronym dictionary* is generated through the reduction to acronyms similar to gene names and removal of long forms containing gene or protein names. In the compilation step all synonyms (also acronyms and their long forms) are classified into one of several classes, which are searched with specific parameter settings like *case sensitive*, *exact* or *permuted* search in the subsequent search queries.

The search system is based on an approximate string matching algorithm enabling not only exact matches but also small variations in spelling. Synonyms which are contained in more than one Entrez Gene entry or additionally found in the acronym dictionary are labeled as ambiguous and the number of different Entrez Gene entries are memorized (D_{occur}). Hits of ambiguous synonyms are only accepted if another unique match (not labeled as ambiguous) of the same entry is found or if the user assigned disambiguation threshold ($D_{\#}$) is higher than the number of different Entrez Gene entries (D_{occur}).

In the training set several protein names are split by the insertion of acronyms put in brackets. As result the full name is not found (coenzyme A (HMG-CoA) synthase). To solve this problem, three runs were made. In the first run the original text was used. For the second run the full bracketed expression was removed and in the third run only the brackets are deleted. The runs are merged and the ProMiner ambiguous filter selects the appropriate matches. In order to disambiguate genes between different organisms the NCBI taxonomy

² Sources: <http://obo.sourceforge.net/>: UniProt taxonomy, Brenda tissue, Mouse adult gross anatomy, Mouse pathology, cellular component, Cell type, DiseaseOntologyV2_1

database [8] is integrated in our system and a simple co-occurrence approach is applied. A gene is rejected from the result set when it is mentioned together with any other organism or different ancestor in the phylogenetic tree than *Homo sapiens*. A relational database system which will be described elsewhere (Dach et al., in preparation) and recursive SQL was used to accomplish this step.

3 Results

Three different runs are computed and submitted, intended to meet highest F-score, precision or recall of the ProMiner system (cf. table 1, bold). Overall, all three runs generate results which position our approach in the first quartile of all participants. The runs differ in the setting for the disambiguation threshold (controlling the result set for matches of gene names which are not unique in the dictionary) and the organism filter. In the first run this threshold was set to one (D1) allowing no matches of non-unique dictionary names. These conditions result in the highest F-measure (0.799). The second run accepts ambiguous matches (D3), increasing recall (+ 0.035) but this is accompanied by a high loss in precision (- 0.054). In the last run we also take the recognition of organism names into account and remove matches in abstracts/sentences only talking about other organisms. This approach leads to a slightly better precision (+ 0.002) but is accompanied by a high loss of recall (- 0.038) and an overall loss in F-measure (-0.02). The original dictionary not adapted to the BioCreAtIvE training corpora demonstrates a loss in precision of 0.024 compared to the final dictionary used in the submitted runs. To show the impact of ambiguity within the dictionary and maximum reachable hits with our dictionary we used a dictionary containing only the gold standard genes. Here precision as well as recall were increased by 0.05. The inclusion of bracket resolution on the training corpus results in an increase of 0.02 in recall but can not be reproduced on the test set. In this case, no differences can be observed between the different runs.

Table 1 ProMiner results

ProMiner runs on the test corpus (Test) with different user assigned disambiguation thresholds (D1, D3), organism selection (O+, O-) were submitted (Run 1-3). The next two columns present results on the test corpus with the originally dictionary without any BioCreAtIvE training (DictOrig) or a dictionary subset containing only gene entities from the gold standard (DictSub). The result on the training corpus is shown in the Train column. The last two columns provide results with a reduced ProMiner run containing no bracket resolution (-brackets) on the training and test corpus.

	Test Run 1 D1, O-	Test Run 2 D3, O-	Test Run 3 D1, O+	Test DictOrig D1, O-	Test DictSub D1, O-	Train D1, O-	Train-brackets D1, O-	Test-brackets D1, O-
F-measure	0.799	0.790	0.779	0.792	0.847	0.784	0.776	0.799
Recall	0.768	0.803	0.730	0.777	0.811	0.755	0.736	0.768
Precision	0.833	0.779	0.835	0.809	0.885	0.819	0.820	0.833
Quartile	1	1	1					

References

- [1] Bairoch, A., Apweiler, R., Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N, Yeh LS.: The Universal Protein Resource (UniProt). *Nucleic Acids Res.*, 33: D154-159, 2005.
- [2] Hanisch D, Fluck J, Mevissen H, Zimmer R: Playing Biology's Name Game: Identifying Protein Names in Scientific Text. *Pacific Symposium on Biocomputing*, 8:403–414 2003.
- [3] Hanisch, D., Fundel, K., Mevissen, H., Zimmer, R., Fluck, J.: ProMiner: Rule based protein and gene entity recognition. *BMC Bioinformatics*, 6(Suppl 1):S14, 2005.
- [4] Schwartz AS, Hearst MA: Identifying Abbreviation Definitions in Biomedical Text. *Pacific Symposium on Biocomputing*, 451-462, 2003.
- [5] Karopka T, Fluck J, Mevissen HT, Glass A.: The Autoimmune Disease Database: a dynamically compiled literature-derived database. *BMC Bioinformatics*, 7:325, 2006.
- [6] Kersey P. J., Duarte J., Williams A., Karavidopoulou Y., Birney E., Apweiler R.: The International Protein Index: An integrated database for proteomics experiments. *Proteomics*, 4(7): 1985-1988, 2004.
- [7] Maglott, D., Ostell, J., Pruitt, K.D., Tatusova, T.: Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Res.*, 1;33:D54-8, 2005.
- [8] Wheeler DL, Chappey C, Lash AE, Leipe DD, Madden TL, Schuler GD, Tatusova TA, Rapp BA (2000). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, 1;28(1):10-4, 2000.



Human Gene Normalization by an Integrated Approach including Abbreviation Resolution and Disambiguation

Katrin Fundel

katrin.fundel@bio.ifi.lmu.de ralf.zimmer@bio.ifi.lmu.de

LFE Praktische Informatik und Bioinformatik, LMU München,
Amalienstr. 17, D-80333 München

Abstract

We present an integrated system for named entity identification and the results of its application for human gene name normalization. The system builds on extensively curated synonym dictionaries and expands on exact text matching and ProMiner by implementing new modules for abbreviation resolution and disambiguation; it achieved encouraging results in the BioCreAtIvE challenge.

1 Introduction

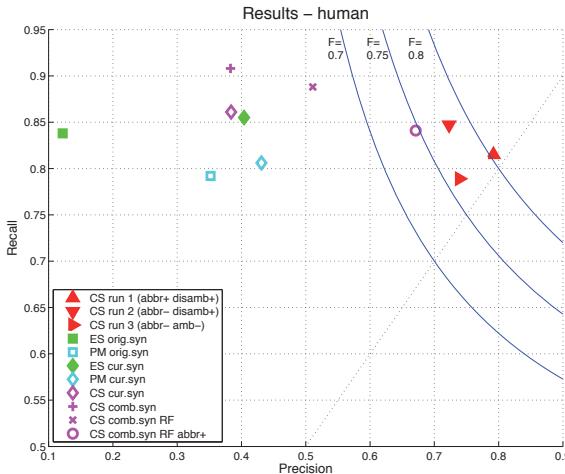
We present an update of the dictionary based approaches applied in the first BioCreAtIvE Challenge [3, 6]. Here, we combine the matching results of the two approaches and focus on post-matching steps: We significantly extended the rule-based post-filter and introduce a method for sense resolution of gene names that overlap with abbreviations or non-gene concepts and gene name disambiguation.

2 Methods

Gene name detection We compiled and automatically curated [3, 5] a large synonym dictionary for human gene names (comb.syn) from Entrez Gene [7], SWISSPROT [1], and HUGO [11]. The curation was tuned towards recall, e.g. by setting the minimum length for a synonym to two characters and by allowing synonyms consisting of a single letter and a number. The resulting dictionary contains 587250 distinct synonyms for the 32969 genes, compared to 168805 synonyms in the original dictionary (normalized, i.e. case-insensitive and ignoring non-alphanumeric characters). Gene and protein names were identified in the texts by exact matching (EM) [3] and ProMiner (PM) [5, 6], a tool that makes use of approximate string matching, and merged into one set of matches (CS).

Rule-based postfilter (RF) A match is pruned if an organism other than human precedes it; terms such as pathway, binding site, region, domain, cell, family, related, syndrome, disorder occur nearby; the synonym consists of a *p* or *q* followed by a number and a term such as chromosome, region, band, deletion, insertion occurs nearby or the match indicated chromosomal context (e.g. *6p21.3-p22*); the synonym resembles a chemical element and the match is followed by '+' or '-'; the synonym is similar to the three-letter code of an amino acid and another three-letter-coded amino acid is found; the synonym resembles a sequence of one-letter code amino acids and one of the amino acids is found in three-letter code or full name; etc. Enumerations are resolved for synonyms that end on a roman or arabic number or single Latin or Greek letter and are followed by further similar specifiers.

Abbreviation resolution and disambiguation Short form/long form pairs of abbreviations were extracted by a rule-based approach. The resulting abbreviation dictionary was combined with a public abbreviation dictionary [4] and all non-gene and non-protein concepts of UMLS [2]. The dictionary entries were represented as feature vectors with word-stems [9] or 3-grams (i.e. all substrings



Parameters	R	P	F
1: $abbr_+ disamb_+$	0.815	0.792	0.804
2: $abbr_- disamb_+$	0.847	0.723	0.780
3: $abbr_- amb_-$	0.789	0.739	0.763

Figure 1: BioCreAtIvE II results for human gene normalization. ES: exact search, PM: ProMiner, CS: combined results from exact search and ProMiner, orig.syn: original synonym dictionary as provided by the organizers, cur.syn: curated original synonym dictionary, comb.syn: curated combined synonym dictionary derived from HUGO, SWISSPROT and Entrez Gene, RF: rule-based filter, $abbr_+$: abbreviation resolution, $disamb_+$: disambiguation, amb_- : ambiguous synonyms pruned from dictionary.

of length 3) as features and inverse document frequency as weights. The entries that do not surpass a certain cosine similarity with any of the gene name dictionary entries are gathered in a dictionary of 'alternative concepts'. Gene names that overlap with short forms contained in the dictionary of alternative concepts are in most cases abbreviations. Therefore, we refer to the disambiguation of gene names versus alternative concepts as *abbreviation resolution* (*abbr*). Gene names that are ambiguous for different genes are subjected to *disambiguation* (*disamb*). We apply the same approach for abbreviation resolution and disambiguation: we determine the cosine similarity between all noun phrase chunks [10, 8] from a given abstract and alternative synonyms of the possible genes/concepts. Then, the object yielding the maximum similarity is reported, provided this similarity is achieved by only one gene/concept and is above a certain threshold (here: 0.5).

Submissions Our three submissions combine matching results of exact search and ProMiner and employ the rule-based post filter (CS comb.syn RF). They differ in subsequent post-processing steps:

- Run 1 ($abbr_+ disamb_+$) implies abbreviation resolution and disambiguation. This run evaluates the full pipeline.
- Run 2 ($abbr_- disamb_+$) implies disambiguation, but no abbreviation resolution. The comparison against run 1 evaluates the relevance and performance of abbreviation resolution.
- Run 3 ($abbr_- amb_-$) implies no abbreviation resolution and no disambiguation. Terms ambiguous with the dictionary of alternative concepts are left in the result set, ambiguous gene names are pruned from the result set. This run marks the baseline.

Furthermore, the individual components of our system were evaluated in several post-evaluation runs.

3 Results and Discussion

The BioCreAtIvE results (Figure 1) indicate good performance of our integrated approach. Application of abbreviation resolution and disambiguation (Run 1) compared to no abbreviation resolution and ignoring ambiguous synonyms (Run 3) results in an increase in Precision, Recall, and F-measure (5.3, 2.6, 4.1 percentage points, respectively). Run 2 yielded highest recall of the submitted runs, while, compared to Run 1 and 3, precision is decreased.

The combined synonym dictionary contains 24 718 ambiguous synonyms, 10 308 entries overlap with abbreviations. For the test set, 671 synonym matches were neither abbreviations nor ambiguous and thus directly accepted, 329 synonym matches were subjected to abbreviation resolution which retained

179 matches, and 128 matches were subjected to disambiguation which retained 81 matches. The synonym dictionary has an important effect on the overall result: The original synonym dictionaries provided by the organizers yield low precision (ES orig.syn, PM orig.syn, 12-35%). Curation leads to a slight increase in recall and an important increase in precision (cur.syn). Compared to the curated original dictionary (CS cur.syn, R:86%), the combined dictionary leads to significantly higher recall (CS comb.syn, 91%) at similar precision (38%). The rule-based filter improves precision (CS comb.syn RF, 51%) at slightly decreased recall (89%). Applying abbreviation resolution for deciding whether a term refers to a gene or alternative concept and reporting all ambiguous synonyms (CS comb.syn RF abbr₊) further improves precision (67%) and decreases recall (84%). Similarly, ignoring ambiguous synonyms (Run 3) leads to increased precision, i.e. an important fraction of the false positives passing the rule based filter are ambiguous synonyms. Our proposed disambiguation procedure yields precision close to using only unique synonyms (72% in run 2 vs. 74% in run 3), but significantly higher recall (85% vs. 79%). Together, the results confirm that abbreviation resolution and disambiguation indeed play an important role in gene normalization.

4 Conclusions

The described system achieved good performance in the BioCreAtIvE challenge. Given that many gene names are ambiguous and overlap with non-gene terms and abbreviations, disambiguation plays an important role in gene normalization. Here, we applied a dictionary-based approach for context-dependent abbreviation resolution and gene name disambiguation. Importantly, this approach relies solely on the information contained in the gene name and alternative concept dictionaries. Thus, it does not require annotated training data which is labor intensive to generate for each ambiguous term; yet, it achieves competitive results.

References

- [1] A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, et al. The Universal Protein Resource (UniProt). *Nucleic Acids Res*, 33 Database Issue:D154–9, 2005.
- [2] O. Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res*, 32 Database issue:D267–70, 2004.
- [3] K. Fundel, D. Guttlér, R. Zimmer, and J. Apostolakis. A simple approach for protein name identification: prospects and limits. *BMC Bioinformatics*, 6 Suppl 1:S15, 2005.
- [4] S. Gaudan, H. Kirsch, and D. Rebholz-Schuhmann. Resolving abbreviations to their senses in Medline. *Bioinformatics*, 21(18):3658–64, 2005.
- [5] D. Hanisch, J. Fluck, H. T. Mevissen, and R. Zimmer. Playing biology’s name game: identifying protein names in scientific text. *Pac Symp Biocomput*, pages 403–14, 2003.
- [6] D. Hanisch, K. Fundel, H. T. Mevissen, R. Zimmer, and J. Fluck. ProMiner: rule-based protein and gene entity recognition. *BMC Bioinformatics*, 6 Suppl 1:S14, 2005.
- [7] D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Res*, 33 Database Issue:D54–8, 2005.
- [8] G. Ngai and R. Florian. Transformation-Based Learning in the Fast Lane. *Proceedings of the North American Chapter of the Association for Computational Linguistics on Language Technologies 2001 NAACL ’01*, pages 40–47, 2001.
- [9] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- [10] L. Smith, T. Rindflesch, and W. J. Wilbur. MedPost: a part-of-speech tagger for bioMedical text. *Bioinformatics*, 20(14):2320–1, 2004.
- [11] H. M. Wain, M. J. Lush, F. Ducluzeau, V. K. Khodiyar, and S. Povey. Genew: the Human Gene Nomenclature Database, 2004 updates. *Nucleic Acids Res*, 32 Database issue:D255–7, 2004.



A Hybrid Gene Normalization approach with capability of disambiguation

Jung-Hsien Chiang¹
jchiang@mail.ncku.edu.tw

Heng-Hui Liu¹
liuhh@cad.csie.ncku.edu.tw

¹Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan

Abstract

Gene normalization is critical for precise biomedical information extraction. We have developed an automatic gene normalization process that takes the output of named entities recognition (NER) systems designed to identify gene mentions and normalizes them to *Entrez Gene IDs*. Most gene mentions referring unambiguously to a unique identifier can be normalized using a thesaurus based procedure by morphological normalization rules. For the rest mentions associated with more than one definition, we propose a hybrid information fusion framework to deal with the ambiguities. An acceptable performance (precision 0.8 and Recall 0.74) was evaluated on 261 articles that BioCreative 2006 provided for training.

Keywords: fuzzy aggregation, gene normalization, maximum entropy classifier, disambiguation

1 Introduction

Gene and protein name identification and recognition in biomedical literature are the earliest steps in information extraction, and performance on these aspects impacts all subsequent steps of the system. In general the process consists of three stages: recognizing named entities in text, identifying the semantic intent of each recognized mention, and normalizing mentions by associating each mention with a gene identifier [2]. There are different challenges in each stage. For last two stages, also known as gene normalization, there are several approaches have been proposed, including classification techniques [1], text matching with dictionaries [2], and combinations of these approaches. The common difficulties are how to construct a comprehensive dictionary, how to match mentions in text to entities in dictionary as morphological variation exists, and how to deal with ambiguities when a mention associated with multiple referents. In our work, we design a system with two-tier architecture to automate the process of gene normalization integrating rule-based approach, maximum entropy classification and fuzzy information fusion techniques. The system is described as following.

2 Method and Results

2.1 System overview

There are two conditions while we match a gene mention in dictionary for their identities (*Entrez Gene ID*) — matches associating a mention with a unique gene id or with more than one ids. The latter causes called ambiguous. We hence developed a two-tiered architecture — one for non-ambiguous mentions and the other for the ambiguous. The system architecture is

as shown in Figure 1. In tier-one, we took advantage of NER module of Lingpipe (<http://www.alias-i.com/lingpipe/>) to identify gene mentions in text. For dealing with the morphological variation, those mentions and entities in dictionary were normalized by normalizing rules before matching. If a normalized mention could be matched in dictionary without ambiguity, we report the corresponding *gene id*. Otherwise, the PMID-normalized mention pair would be passed to tier-two for disambiguation. In tier-two a trained maximum entropy classifier plays the role of an expert that provides different levels of confidence for each candidate *gene id* according to features extracted from context of the abstract. Information derived from classifiers would then be fused into a single fuzzy set by an aggregation operator, and the top candidate *gene id* with membership degree greater than certain threshold would be selected as the referent. Details of each part of the system would be described in following sessions.

2.2 Tier-one : thesaurus-based normalizer

Although costly to compile, an accurate gene name dictionary with sufficient coverage is an essential piece of a gene normalization system. Fortunately, there have been several previous efforts in this area such as BioThesaurus[4], which is designed to map a comprehensive collection of protein and gene names to protein entries in the *UniProt* Knowledgebase. We collected 230,000 gene/protein names of human from BioThesaurus as basis of our dictionary. To increase recall related to morphological variation, five normalization rules were considered while comparing output of NER and name entities in dictionary. The rules include normalization of case, replacement of hyphens with spaces, removal of all spaces, removal of punctuation, and removal of parenthesized materials.

2.3 Tier-two: hybrid information fusion framework for disambiguation

If an ambiguous match occurs in tier-one, we are not able to correctly identify a mention using morphological features alone. More wide and deep analysis should be considered to increase system performance. Contextual information may provide some clues to identities of mentions and could be used as features for classification [1]. In this application, MeSH headings, gene mentions, GO terms etc. and their combinations are considered as features. For example, one of these features is

$$f(\mathbf{x}, y) = \begin{cases} 1 & \text{if 'apoptosis' and 'tumor' in } \mathbf{x} \text{ & mention} = 'tp53' \text{ & guessID} = 7157 \text{ & } y = \text{yes} \\ 0 & \text{otherwise} \end{cases}$$

Here \mathbf{x} is context of an abstract, and y is a class. Maximum entropy will construct a stochastic model faithfully from the training data without any assumption on relationships of features. A trained classifier would assign the probability $p(y|\mathbf{x})$ to y in context \mathbf{x} . For an ambiguous match, a mention would be associated with multiple gene ids so we used a fuzzy set to

represent the ambiguous match. If a mention associates with three ids, for example, it could be represented as

$$\frac{T_x^{id_1}}{id_1} + \frac{T_x^{id_2}}{id_2} + \frac{T_x^{id_3}}{id_3},$$

where $T_x^{id_i}$ is membership degree of which id_i belongs to the mention in context x . A function T will convert $p(y|x)$ s of a classifier to corresponding membership degrees. We trained two maximum entropy classifiers according different feature types to derive the required fuzzy sets. To integrate information from different classifiers, fuzzy sets derived from classifiers would be merged by ordered weighted averaging (OWA) operator [3] into single fuzzy set where a membership of certain candidate id represents level of confidence of which the mention refers to. Finally, candidate ids whose degrees were higher than a threshold would be reported.

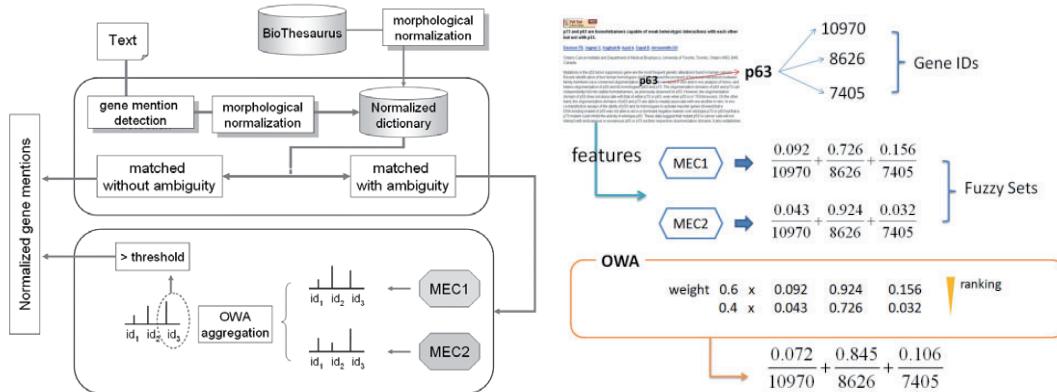


Figure 1: System architecture and an example. The upper block is tier-one, which normalizes gene mentions without ambiguity and morphological normalization rules are adopted here for string matching. MECs in lower block are Maximum Entropy Classifiers that provide degree of confidence of candidate ids for ambiguous gene mentions.

References

- [1] Crim, J., McDonald, R., Pereira, F., Automatically annotating documents with normalized gene lists, *BMC Bioinformatics*, 6: S31, 2005
- [2] Fang, H.-R., Murphy, K., Jin, Y., Kim, J. S., White, P. S., Human gene name normalization using text matching with automatically extracted synonym dictionaries, *Proc. BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL 06*, 41-48, 2006
- [3] Klir, G. J., Yuan, B., *Fuzzy Sets and Fuzzy Logic: theory and application*, Prentice Hall Press, 2003
- [4] Liu, H., Hu, Z. Z., Zhang, J., Wu, C., BioThesaurus: a web-based thesaurus of protein and gene names, *Bioinformatics*, 22(1): 103-105, 2005



Exploring Match Scores to Boost Precision of Gene Normalization

Cheng-Ju Kuo¹

cju.kuo@gmail.com

Yu-Ming Chang²

porter@iis.sinica.edu.tw

Han-Shen Huang²

hanshen@iis.sinica.edu.tw

Kuan-Ting Lin²

woody@iis.sinica.edu.tw

Bo-Hou Yang^{2,3}

ericyang@iis.sinica.edu.tw

Yu-Shi Lin²

bathroom@iis.sinica.edu.tw

Chun-Nan Hsu²

chunnan@iis.sinica.edu.tw

I-Fang Chung¹

ifchung@ym.edu.tw

¹ Institute of Bioinformatics, National Yang-Ming University, Taipei, Taiwan

² Institute of Information Science, Academia Sinica, Taipei, Taiwan

³ Department of Electrical Engineering, Chang-Gang University, Tao Yuan, Taiwan

1 Introduction

Gene normalization task is to identify EntrezGene IDs corresponding to the human genes and direct gene products appearing in a given MEDLINE abstract. Given a dictionary that maps gene and protein synonyms to EntrezGene IDs, a naive approach to the problem is to apply a gene mention tagger to identify all potential name entities of genes and then look them up in the dictionary. However, mostly due to the difficulty to compile a complete yet noise-free dictionary for gene synonyms [5], the results are far from satisfactory. In our experiments using a gene mention tagger based on a conditional random field (CRF) [4] model and a string matcher based on softTFIDF [1] to look up the dictionary, the F-score is below 0.5. To improve the performance, previous work proposed many methods to clean up dictionaries. These methods may help case by case but may not applicable in general. In this paper, we focus on the problem of whether there exists a systematic method that always improves the result of dictionary lookup. We propose to train an ensemble of classifiers using AdaBoost [2] to recognize true positives from false ones based on the match scores, which are readily available when anyone applies an approximate string matching function to look up the dictionary. Experimental results show that applying boosting can successfully increase the F-score from about 0.56 to 0.69 with our best F-score reaching 0.75. These results were obtained without modifying the dictionary.

2 Method and Results

Given an abstract, our system takes the following three steps to return the EntrezGene IDs mentioned in the abstract:

1. We directly apply our gene mention tagger from the GM Task [3] to identify possible entities of gene names.
2. For each entity, we apply an approximate string matching function to compare the entity with all entries in the dictionary. Each entry contains the EntrezGene ID and the synonyms of a human gene. A list of top ten match scores is returned with the ID of the top match.
3. Based on the match scores, an ensemble of classifiers is applied to determine if the top ID actually corresponds to the entity. If positive, the ID with its score will be returned; otherwise, the result will be discarded.

We describe the details of these steps as follows. Our gene mention tagger is a union of bidirectional parsing CRF models from the GM Task [3]. This tagger was trained by the 15,000 training examples from the GM Task, which contains data of gene/protein names of all species. But we only need human gene/protein names in the GN Task. As a result, though this tagger achieves a 0.8658 F-score for the GM Task, its F-score for the training data of the GN Task is far from this level. However, since it is not necessary to identify all gene mentions in abstracts, before a training data set for human genes is available, we will have to settle with this gene tagger.

Next, we applied TFIDF and softTFIDF [1] to compute the similarity between a tagged gene name entity and a synonym in the dictionary. A preprocessing step that transforms a string into a token vector was applied in advance when we applied TFIDF, including case normalization, replacement of hyphens with blanks, removal of punctuation symbols and parenthesized strings, etc. The idea is to increase the chance of matchings. For softTFIDF, there is no need to perform the preprocessing step because we have Jaro and Jaro-Winkler with TFIDF to tolerate slight difference between terms in gene names. We assigned a threshold δ to filter the outputs of the dictionary lookup. If the highest match score is less than δ , the tagged entity in the abstract will be discarded. Otherwise, the ID with the top score will be returned as an answer. This is how we obtained the results of Step 2 in Table 1.

The feature vector for our ensemble classifier is derived from the top ten match scores of synonyms of ten distinct genes. Let $(s_1, s_2, \dots, s_{10})$ be the top ten scores, the feature vector consists of twelve features defined as follows:

$$(s_1, s_1 - s_2, s_2 - s_3, \dots, s_9 - s_{10}, s_1 - s_{10}, \text{Var}(s_i)).$$

The idea is to characterize the distribution of the match scores to discriminate a false positive. This feature set assumes that the dictionary contains entries that share many terms such that an entity in an abstract may match many synonyms in the dictionary. We applied AdaBoost to train an ensemble classifier with this feature set because boosting can automatically take advantage of the fact that these features are not equally important. We stopped iterations of AdaBoost at thirty because ensembles with thirty decision stumps performed the best in our experiments. Suppose the accuracy of our classifier is α , then the new F-score after applying our classifier will be:

$$P = \frac{TP \cdot \alpha}{TP \cdot \alpha + FP(1 - \alpha)}, \quad R = \frac{TP \cdot \alpha}{TP + FN}, \quad F = \frac{2PR}{P + R}.$$

Therefore, if we have a dictionary lookup result whose FN is small but TP and FP are large (i.e., low precision and high recall), then our classification method will boost the precision as well as the F-score.

When more than one entry in the dictionary share a top match score, our feature set would be insufficient to recognize which entry is a true positive. In this case, we have two tie-breaking (TB) strategies to handle the situation. One is to simply discard that entity to reduce false positives. The other is to return the ID of the entry that maximizes the occurrences that the entity appears as a substring in the synonyms of that entry. The rest will be sent to the classifier for further filtering.

Table 1 shows the results of our experiments. All trials used the output of our CRF tagger as the input. Due to the time constraint, we only had the result of the configuration – TFIDF and $\delta = 0.5$ before the deadline. Apparently, the threshold is too low so that Step 2 passed many false positives to Step 3. However, our classifier still successfully filtered most of them to improve the F-score from 0.56 to 0.69. After the deadline, we raised the threshold to decrease false positives by Step 2 and the F-scores went up. In our experiments with softTFIDF, since approximate string matching was used to compute the similarity, the scores are usually higher than TFIDF. Therefore, higher thresholds are necessary for softTFIDF. We found that the F-score was increased as we increased δ but when $\delta = 0.99$, Step 3 failed to boost the F-score because in these cases, it is recall that needs boosting rather than precision. Nevertheless, our best F-score was achieved when Step 3 was applied to boost Step 2 with $\delta = 0.95$ and our tie-breaker applied.

Table 1: Performance Comparison for Gene Normalization.

Method	δ	Step2 (Precision/Recall/F-score)		Step3
		Step2	Step3	Step3
TFIDF	0.5	0.4523/0.7375/0.5607	0.7166/0.6636/0.6891(* submitted)	
	0.9	0.6495/0.6777/0.6633	0.8402/0.6229/0.7154	
	0.95	0.6904/0.6714/0.6714	0.8637/0.5974/0.7063	
softTFIDF (Jaro)	0.9	0.6163/0.7286/0.6678	0.8155/0.6700/0.7356	
	0.95	0.7496/0.6866/0.7167	0.8484/0.6560/0.7399	
	0.99	0.8210/0.6547/0.7285	0.8539/0.6331/0.7271	
softTFIDF (Jaro-Winkler)	0.9	0.4670/0.7503/0.5757	0.7524/0.6968/0.7235	
	0.95	0.6389/0.7235/0.6786	0.7957/0.6751/0.7305	
	0.99	0.8077/0.6636/0.7286	0.8341/0.6407/0.7247	
softTFIDF+TB (Jaro)	0.9	0.5907/0.7630/0.6659	0.7890/0.7006/0.7422	
	0.95	0.7159/0.7222/0.7209	0.8256/0.6878/0.7505	
	0.99	0.7918/0.6929/0.7391	0.8328/0.6662/0.7402	
softTFIDF+TB (Jaro-Winkler)	0.9	0.4555/0.7834/0.5761	0.7315/0.7324/0.7320	
	0.95	0.6172/0.7579/0.6803	0.7699/0.7121/0.7399	
	0.99	0.7779/0.7006/0.7372	0.8027/0.6789/0.7356	

References

- [1] Cohen, W. W., Ravikumar, P., and Fienberg, S. E. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration of the Web*, 2003.
- [2] Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.
- [3] Kuo, C.-J., Chang, Y.-M., Huang, H.-S., Lin, K.-T., Yang, B.-H., Lin, Y.-S., Hsu, C.-N., and Chung, I.-F. Rich feature set, unification of bidirectional parsing and dictionary filtering for high f-score gene mention tagging. Submitted to Second BioCreAtIVe Challenge Workshop, 2007.
- [4] Lafferty, J., McCallum, A., and Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
- [5] Morgan, A. A., Wellner, B., Colombe, J. B., Arens, R., Colosimo, M. E., and Hirschman, L. Evaluating the automatic mapping of human gene and protein mentions to unique identifiers. In *Pacific Symposium on Biocomputing*, pages 281–291, 2007.



Rule-based Gene Normalization with a Statistical and Heuristic Confidence Measure

William Lau¹
`william.lau@nih.gov`

Calvin Johnson¹
`johson@mail.nih.gov`

¹ High Performance Computing and Informatics Office, Division of Computational Biosciences, Center for Information Technology, National Institutes of Health, Bethesda, Maryland, United States

Abstract

In the gene normalization task, a rule-based approach has certain advantages including the fact that no gold standard is likely to contain all the genes that need to be considered. We have developed a rule-based algorithm that includes pattern matching for gene symbols and an approximate term searching technique for gene names. The algorithm performs confidence estimation by appropriately weighting measures of uniqueness, inverse distance, and coverage. An F-measure of 0.753 has been achieved, using nominal confidence-measure weights.

Keywords: gene normalization, rule-based, approximate term search, confidence measure

1 Introduction

The gene normalization algorithm we entered to Task 2 of the second BioCreAtIvE challenge is a prototype component of a text mining tool for genetic association studies [1]. The goal of this tool is to provide a means to systematically identify associations between sets of genes and diseases using information available in the MEDLINE literature. The assumption is that if the co-occurrence frequency between a gene and a disease is of statistical significance, they probably have an underlying biological relationship. Since simple string matching of the genes has yielded poor performance [3], we developed the gene normalization algorithm presented in this paper.

Several elements were taken into consideration when we designed the approach we employ herein. We chose a rule-based approach since many genes that we anticipated to encounter in the citations were not in the gold standard provided to us. We can tolerate a few random errors as they would not likely influence the association results. Given these factors, the algorithm we have developed is a balancing act between simplicity, accuracy, and computational efficiency.

2 Methods

2.1 Identification of Gene Mentions

The algorithm detects the occurrence of gene mentions by matching input text against an EntrezGene dictionary. The procedure effectively combines the tasks of gene detection and gene identifier lookup. Instead of using the lexicon provided to us, we created our own knowledge base with more comprehensive synonyms. Different approaches were used in the detection for gene symbols (including “Other Aliases” in the EntrezGene database) and the detection of gene names (including “Other Designations”). Gene symbol tagging is based on pattern matching. A set of regular expressions rules are applied to evaluate every string separated by space and punctuation symbols. The rules are commonly used in gene recognition tasks to account for syntactic variations, such as the interchange of Roman and Arabic numerals, placement of dashes

* This research was supported by the Intramural Research Program of the NIH, Center for Information Technology.

and spaces, case difference and plurality. For the official symbols, we also generate new symbols by expanding the associated Greek letters into their full names, e.g. “CHKB” to “CHK beta” and “beta CHK”.

For gene names, an approximate term matching technique is employed. After breaking a gene name into individual tokens, each token is searched against the text. Subsequently, the phrase containing the most tokens is identified. The ratio, r_m , between the number of tokens in the mention candidate and the total number of tokens needs to be higher than a threshold (0.7 in our submissions) for the phrase to be accepted. However, the candidate has to include specific tokens as measured by the number of citations containing those tokens (if a token’s frequency of occurrence is low, it is too important to be ignored). The system also maintains a list of allowed and prohibited missing words. If a word in the prohibited list, e.g. “receptor”, is missing from the phrase, the candidate is rejected. On the other hand, if a word in the allowed list, such as “type” and “subunit”, is missed, the algorithm calculates r_m as if the word were not in the gene name. In addition, candidates are allowed to contain at most two extra words between any two tokens providing that the words are frequently found in the biomedical literature. Besides the names that are already in the knowledge base, additional synonyms are generated by replacing common chemical names with their abbreviations. This approximate matching technique, which is similar to that proposed by Hanisch *et al* [2], can accommodate typical variations of gene name mentions, such as word ordering, found in the literature.

2.2 Confidence Measure of Gene Mention Candidates

After a gene mention is detected, the algorithm calculates a confidence score using several statistical and heuristic measures. The three main factors used in our submissions were *uniqueness*, *inverse distance*, and *coverage*. Each of them contributed to 20% of the confidence score:

1. *Uniqueness* is an estimate of the probability that the candidate is referring to something other than the gene in question. If the mention has a very high frequency of occurrence in the literature, the score is reduced accordingly, because frequently occurring terms may have multiple meanings other than just being referred as genes.
2. For gene symbols, *inverse distance* is based on the edit distance of the candidate term to the formal reference in the database. It takes into consideration the variations in capitalization, ordering, and any omissions/additions of punctuations and spaces. The closer the mention matches the actual symbol, the higher the score. For gene names, since syntactic variations are common, the inverse distance is the harmonic mean of edit distance and token ratio r_m .
3. The calculation of the *coverage* score is quite different between gene names and gene symbols. For symbols, this score is calculated as follows:

$$\psi_s = \left(\frac{\tan^{-1}(2L - 3)}{\pi} + 0.5 \right) \times s$$

where L is the symbol character length and, s is a scaling factor defined as:

$$s = \begin{cases} \left(e^{\frac{r_m - 1}{L}} \right)^2 & \text{if the candidate is enclosed} \\ 0.8 + 0.2e^{r_m - 1} & \text{otherwise} \end{cases}$$

The intuition is that the more characters the symbol has, the less likely it is that the term is used other than to represent the gene. If the term is enclosed by brackets, i.e. ([{}]), the gene name is probably mentioned in the text as well and score should be scaled accordingly.

For gene names, coverage is a weighted average of two ratios, r_l and r_m . r_l is the ratio between the character length of the candidate string and the corresponding name in the knowledge base. Thus,

$$\psi_N = \frac{1}{2} r_m \left(\frac{3f_{th} + 1}{f_m} \right) + \frac{1}{2} r_l$$

where f_{th} is the minimum occurrence frequency threshold for any missing words not in the allowed list (set to 20,000), and f_m is the occurrence frequency of the least common missing word. In addition to character length, the coverage metric for gene names also takes into account how many words are matched as well as the specificity of the words missing from the mention.

With 10% of the score reserved for future features, the remaining 30% of the confidence score was calculated using the factors listed in Table 1. Furthermore, we incorporated a boosting factor to reward or punish the candidate when there was other evidence in the text to suggest whether the mention actually referred to a gene. For example, if the text contained the chromosome location of the gene, its score would be boosted. If the mention was preceded or followed by supporting modifiers, such as “gene” and “encode”, our level of confidence would increase. On the contrary, if counter-indicators, such as “test” and “cell line”, appeared adjacent to the candidate, the score would drop. Whereas all the other factors were combined linearly to compute the final score, the boosting factor was added last as an exponent to the score.

Table 1: Factors used to calculate 30% of the confidence score when a gene mention is detected.

Factors	Contribution
Whether the mention is an official gene term?	18%
Whether more than one mention is detected for the gene?	10%
Whether the gene is approved by the HUGO Gene Nomenclature Committee?	2%

2.2 Disambiguation

When a string is associated with more than one gene identifier, the algorithm needs to determine which gene the authors actually intended. The disambiguation procedure is as follows. First, if a mention appears entirely within another longer mention, the algorithm removes the shorter mention. If some words of a mention overlap with another mention or if two mentions share the exact same term, the one with the lower score is removed. If the scores of two conflicting candidates are equal, their uniqueness scores are both reduced by half. If the candidate had more than one form of occurrence, e.g. both the symbol and the name were detected, the highest score was considered.

3 Analysis

Table 2 shows the performance of our three submissions to the competition. Only candidates with a confidence score higher than a threshold (t_a) were accepted. Since the time of the submissions, we fixed several minor bugs in the system, and an F-measure of 0.753 ($t_a = 0.65$) was achieved on the same set of data. We are currently investigating optimal weights for the various confidence measures, as opposed to the equal (0.2) weighting used here. Preliminary evidence suggests that a greater weight is appropriate for the uniqueness measure followed by the coverage score, and that the inverse distance weight should be reduced.

Table 2: Results of the submissions generated with different acceptance thresholds of the confidence scores.

Run	Threshold (t_a)	Precision	Recall	F-measure
1	0.6	0.655	0.796	0.719
2	0.625	0.690	0.782	0.733
3	0.65	0.726	0.749	0.737

References

- [1] Becker, K.G., Hosack, D.A., Dennis, G.Jr, Lempicki, R.A., Bright, T.J., Cheadle, C., et al., PubMatrix: a tool for multiplex literature mining, *BMC Bioinformatics*, 10(4):61-66, 2003.
 - [2] Hanisch, D., Fundel, K., Mevissen, H.T., Zimmer, R., Fluck, J., ProMiner: rule-based protein and gene entity recognition. *BMC Bioinformatics*, 6(Suppl 1):S14, 2005.
 - [3] Jenssen, T.K., Laegreid, A., Komorowski, J., Hovig, E., A literature network of human genes for high-throughput analysis of gene expression, *Nat.Genet.*, 28(1):21-28, 2001.



Automatically Expanded Dictionaries with Exclusion Rules and Support Vector Machine Text Classifiers: Approaches to the BioCreAtIve 2 GN and PPI-IAS Tasks

Aaron Michael Cohen¹
cohenaa@ohsu.edu

¹ Department of Medical Informatics and Clinical Epidemiology, Oregon Health & Science University, Portland, Oregon, USA

Abstract

For BioCreAtIvE 2 we participated in the gene normalization task (GN) and the protein-protein interaction article subtask (PPI-IAS). Our GN submission used automatically extracted and expanded symbol dictionaries, along with manually generated exclusion rules to filter out likely false positives. Our best submission achieved an F1 of 0.724, which placed it in the second quartile. Our best PPI-IAS submission was a “bag of words” linear SVM system with chi-square based feature selection. This system achieved an AUC of 0.8284, which was greater than one standard deviation above the mean. We were able to improve these results slightly by including all features instead of performing the feature selection step. While our submissions performed well, it is likely that these results can be improved with further study. One particularly interesting question is why cross-validation on the PPI-IAS training set grossly overestimates the results achieved on the test collection.

Keywords: named entity recognition, text classification, machine learning, bioinformatics

1 Introduction

Biological *named entity recognition and normalization* (NER+N), and *text classification* (also called *categorization*) are two machine learning technologies fundamental to biomedical document processing [1]. Good performance on these tasks is an important ingredient in bringing useful automated and computer-assisted text processing systems to working biomedical researchers. The second BioCreAtIvE conference included challenge tasks in both of these two areas.

The BioCreAtIvE conference organizers provided expert-derived training and test collections for a human gene NER+N task, called the GN (gene normalization) task, as well as a gene-gene interaction document classification task. The goal of the human gene NER+N task was to identify the human genes mentioned in each of 262 test documents which consisted of the title and abstract for the corresponding MEDLINE entries. A training collection consisting of 281 similar documents was provided along with a list of the Entrez identifiers for the genes mentioned in these documents.

The text classification task, called the PPI-IAS (protein-protein interaction article subtask), used a document set also derived from MEDLINE, consisting of an XML syntax of a subset of the MEDLINE record data, including title and abstract, for 5495 training documents and 750 testing documents. The training documents included expert-derived decisions on whether or not the document included protein-protein interaction information. The task was to prediction the presence of absence of protein-protein interaction information on the test collection.

2 Gene Normalization Task

2.1 Methods and Results

The human GN for BioCreAtIvE 2 was much like the three species gene normalization tasks conducted for BioCreAtIvE 1, and therefore our approach was similar. Again, we used automatically extracted dictionaries with synonym expansion, separate case-sensitive and case-insensitive matching dictionaries, prefix-optimized exact lookup matching, post-match delimiter detection instead of tokenization, and ambiguity detection and removal [2]. However, some additional specializations were added for the human GN task.

Initially we thought that since many human gene names, synonyms, and symbols (all referred to as symbols here for convenience) appear orthographically similar to that of mouse, using the same system after substituting the human Entrez dictionary entries for the mouse entries would perform similarly. This turned out not to be true, not because human gene symbols are so different from those of mouse, but instead because of the way that the gene name entries are represented in Entrez. For human gene names, but not mouse, much more information is included within the name and symbol entries, with names often consisting of pairs of comma-separated clauses along with parenthetical expressions. These needed special handling in order to extract gene symbols that would be useful in identifying gene symbols that are actually found in the literature.

We downloaded the Entrez gene database on June 6, 2006, extracted out the entries for human genes, and as in our previous work, created a symbol dictionary using extraction and expansion rules on the Entrez database entries. First we extracted out the GENE_ID, SYMBOL, SYONYMS, NAME, and OFFICIAL fields from the human entries (TAXONOMY_ID = 9606) in Entrez database file. We skipped entries marked as “withdrawn”, and removed symbols less than three characters. We expanded the symbols set by subjecting each of these symbols to repeated application of the following rules, until no new symbols were generated:

1. Remove any parenthetical expression containing one or more spaces.
2. If the symbol includes a comma, remove it and reverse the clauses.
3. Split by spaces, remove any of the resulting internal words that only include digits.
4. Remove words corresponding to Greek letter names.
5. Append an “h” to any symbol containing no spaces and less than or equal to 8 characters.
6. Replace spaces with hyphen.
7. Replace hyphen with spaces.
8. Remove hyphen.
9. If a symbol ends with a sequence of digits and possibly a final letter, insert a hyphen before the digits.
10. Replace “-a” and “-b” with “-alpha” and “-beta”.
11. Replace “-i” and “-ii” with “-1” and “-2”.
12. Replace “b1”, “b2”, “b3” with “-beta1”, “-beta2”, “-beta3”.

These rules were determined by inspection and tuning on the training data. As a final step, we removed any resulting dictionary entries less than 3 or longer than 48 characters. Each expanded symbol entry was linked to the corresponding Entrez gene identifier in order to allow normalization as well as recognition. Rules 1, 2, 3, 4, 5, and 12 were added for human gene NER+N, beyond the prior system that we used for mouse genes. As this rule list is much longer than what we found necessary mouse genes, an automated means of extracting these rules, such as that proposed by Tsuruoka becomes more desirable [3]. However, this requires large amounts of training data consisting of sets of synonymous gene symbols. The human gene information in the Entrez database and the BioCreAtIvE 2 GN data are probably not adequate for this, although the Gene Mention (GM) task data may be.

We did some initial experiments combining the gene symbols from the Genew database with those from Entrez, which we found helpful in our previous work. While our previous work with mouse genes found this improved performance, performance on the human training data was somewhat decreased. The results reported here use only gene symbols extracted and generated from information in the Entrez database.

Our system performs post-match delimiter detection instead of tokenization. What this means is that we first search for string matches over the entire text sample, and if a match is found, we check to see whether it is bounded by acceptable delimiting characters. This technique avoids one of the main problems with prior tokenization, which is that it is difficult to allow gene symbols to contain delimiters when tokenization is done up front. The delimiters that we allow by default include the characters space, tab, newline, return, single quote, double quote, slash, backslash, as well as “.,(){}[]=?*!”.

The default delimiters are all single characters. After examining the training data, we decided that it would be worth experimenting with multi-character delimiters. We created two types, *inclusions* and *exclusions*. Inclusions are essentially sequences of characters that are allowed to delimit a gene symbol. The training data yielded two potential candidates “-mediated” and “-induced”. Exclusions are text patterns occurring nearby to the matched pattern that could indicate that the match should not be treated as a found gene mention. By examining errors that our system made on the training data we found 43 exclusion patterns that improved performance on the training data. Some sample exclusion patterns are shown in Table 1.

Our final submission consisted of the output for three variations of our system. Run 1 was our baseline system that included the human-specific dictionary expansion, but no inclusions or exclusions. Run 2 added exclusions, and Run 3 added the two inclusions. Results on the test data for all three runs are shown in Table 2.

2.2 Analysis

Overall performance of all three runs was good, with all three finishing in the second quartile of submitted runs. While the performance of the three systems was comparable, some interesting differences emerged. As expected, adding exclusions to the baseline system improved precision. Recall did not drop very much, and this resulted in our highest performing system, yielding an F-measure of 0.724. This shows that the exclusions were successful in eliminating false positives without creating new false negatives. On the test set, exclusions helped eliminate 16 false positives, and only caused 3 additional false negative errors.

The inclusions however, were not as successful. Paradoxically, the inclusions resulted in a drop of recall and a small increase in precision. This was unexpected, but can be explained by the interaction of the inclusions with the ambiguity removal portion of the main system [2]. If a symbol maps to more than one gene, and neither gene is unambiguously found in the text sample, then no gene identifier for this symbol is output. If the inclusion rules caused an increase in ambiguity, then the result would be an increase in false negatives and a decrease in recall, as was seen on the test collection.

3 Protein-Protein Interaction Article Subtask

3.1 Methods and Results

The data for the PPI-IAS subtask was provided in XML files containing separate records for each referenced journal article. Early in our experiments with the training data we determined that there was a systematic bias in the publication dates for the positive and negative articles. We discussed this with the task director and found out that the publication information (in the SOURCE field) would be blanked out in the test data. Therefore we focused our machine learning work on the TITLE and ABSTRACT fields of the training and test collections, and ignored the publication name, date, etc. From our conversation with the task director we had expected the PMID field to be filled with an obfuscated value, and therefore did not retrieve the rest of the MEDLINE record for use as features (such as the assigned MeSH terms). However, when the test collection arrived, the PMID value was not obfuscated. Nevertheless, all of our experiments and submissions used only the contents of the TITLE and ABSTRACT fields.

Our submissions were based on the support vector machine (SVM) classifier in the SVMLight package [4]. The contents of the TITLE and ABSTRACT fields were tokenized into features using the *StandardAnalyzer* available in the Apache Lucene package. For the submitted runs, we performed feature selection using the chi-squared statistic with an alpha of 0.05. We also performed some subsequent experiments using all

token-based features. Samples were modeled as binary feature vectors. Two variations of SVM-based classifiers were used: linear with default settings, and radial-basis-function (RBF) with grid-search tuned parameters ($C=2.0$, $\gamma=2e-2$). We also tried an idea that we term *output document modeling*, which models each document as a vector of similarities to the documents in the test collection, using the cosine similarity measure. With this method, learning on the training data is based on how closely the training documents appear to be like the unclassified test documents. When the test documents are classified, their feature vectors show that they are exactly like themselves (feature value = 1.0) and somewhat less similar to other documents in the test collection. These test set similarity vectors are then classified using the model derived from the training set similarity vectors.

The results of these system variations on the training and test collections are shown in Table 3. Results on the training collection were obtained using 5x2 cross validation. Results on the test collection are for the single run. Submitted runs are shown as RUN1, RUN2, and RUN3.

3.2 Analysis

Several interesting observations emerge from Table 3. First, the performance of all of our systems was better than the average submitted, often much better. The straight “bag of words” runs using linear and RBF SVM do especially well. The output document modeling technique was not successful, and hurt performance relative to the simpler “bag of words” methods, but still performed better than the average submitted run.

The simple linear SVM with X^2 feature selection was our best performing submission with the highest AUC of 0.8284, while the RBF run had the best F1 score of 0.7649. The differences are small. What is significant was that tuning the RBF run took a lot of cross-validation time performing a grid-based search for the optimal C and gamma parameters. It does not appear that the results justify the increased time or procedural complexity. This result is consistent with other recent comparisons of SVM using a linear kernel versus higher order kernels for biomedical text classification [5]. For biomedical text classification with SVM the linear kernel is sufficient.

Both the F1 and AUC performance improved a bit using the linear SVM with all available features. This is the opposite effect from that we have seen when classifying full text biomedical documents [6]. In previous work we found that better performance could be obtained by using the full article text as opposed to just title and abstract, but aggressive feature selection was necessary to avoid reduced performance caused by overburdening the classification algorithm with too many noisy features [7]. It appears that it is beneficial to include all features when the text is limited to titles and abstracts, but feature selection may be necessary when dealing with full text articles. The difference in optimal classification approaches for full text versus title plus abstract is an interesting area for future study.

Finally, the difference in performance between the cross-validation experiments using the training data and the results on the test collection was dramatic and larger than the difference between any two of our systems run on the test collection. While our initial experiments uncovered a bias in the training data due to the publication date, there may be other undiscovered biases in the training collection that cause over-training and result in decreased performance on the test collection.

In an attempt to understand this, we reversed the roles of the training and test sets, that is, we trained on the 750 classified documents in the test collection, and then applied the resulting model to the 5495 samples in the training set. The result of this experiment is shown in Table 3 as the linear SVM system with the dataset “reverse train/test”. Quite interestingly, the performance of this “reversed” task is much closer to that of the cross-validation experiments on the training set, achieving an AUC of 0.9173. We also performed 5x2 cross-validation using only the 750 samples in the test collection, and obtained the results shown in the line with the dataset “test crossval”. This achieved an AUC of 0.8149 making these results more in line with the results of the submitted entries.

It is unlikely that the disparity of performance could be due to over-fitting on the training collection. The cross-validation results obtained on the training set provides low-bias estimates of the performance obtainable

on samples from the true distribution of the training population, and here these are consistently high [8]. Furthermore, the relatively low performance on the test collection cross-validation indicates that random halves of the test collection did not provide ample information to separate this data with as high accuracy as the training set itself could be separated. Therefore, it does not appear that the training and test collection are good approximations of independent and identically distributed (i.i.d.) samples from the same overall population, one of the fundamental assumptions in most machine learning approaches [9].

These results and observations support the idea that there are features important for accurate classification of the test collection that are sparsely represented within the test collection (probably because it is small) and, more importantly, not represented in the training collection. The test collection however, seems to include sufficient feature information to accurately classify the training data. Therefore it is likely that the test collection includes articles on some topics not present in the training set. This situation may have arisen if the training and test collections were assembled from documents written in widely separated years. While the test data released to the task participants did not include this information, it is certainly easily available to the task administrators. If the publication year histograms of the training and test collections are very different, the results obtained from this task may be a significant underestimate of the results potentially obtainable in the real world task that the PPI-IAS was intended to model.

4 Conclusions

Our approaches to the GN and PPI-IAS tasks performed well above the median and mean submissions, respectively. However, it is unlikely that this is the best performance that can be achieved on these tasks. For the GN task, many questions remain about the best way to use curated database information to create a human gene symbol dictionary, and how to integrate dictionary and machine-learning based NER+N approaches to maximize performance. For the PPI-IAS task it remains to be tested whether integrating metadata such as MeSH assignments could improve performance. Furthermore it is unclear whether the PPI-IAS training collection is really representative of the data in the test collection and whether the results accurately reflect the performance achievable on this task.

5 Tables

Table 1: Examples of exclusion patterns used in GN task.

mouse <GENE SYMBOL>	Putative <GENE SYMBOL>	<GENE SYMBOL> receptor-associated
murine <GENE SYMBOL>	(<GENE SYMBOL>)-related	<GENE SYMBOL> family
rat <GENE SYMBOL>	syndrome <GENE SYMBOL>	<GENE SYMBOL> domain

Table 2: Results of GN task for three submitted runs.

	Run1 Baseline System	Run2 add exclusions	Run3 add inclusions
Recall	0.707	0.703	0.699
Precision	0.731	0.746	0.749
F-measure	0.719	0.724	0.723
True Positives	555	552	549
False Positives	204	188	184
False Negatives	230	233	236

Table 3: PPI-IAS task performance of submitted runs and other experiments.

System	Dataset	Precision	Recall	F1	AUC
Linear SVM, X^2 feature selection	train crossval	0.9380	0.9430	0.9410	0.9721
RBF SVM, X^2 feature selection	train crossval	0.9372	0.9472	0.9422	0.9727
Linear SVM, all features	train crossval	0.9398	0.9463	0.9430	0.9736
Linear SVM, X^2 feature selection (RUN1)	test	0.6808	0.8587	0.7594	0.8284
Linear SVM, output document modeling (RUN2)	test	0.6673	0.8773	0.7581	0.7928
RBF SVM, X^2 feature selection (RUN3)	test	0.6813	0.8720	0.7649	0.8271
Linear SVM, all features	test	0.6864	0.8640	0.7651	0.8325
Mean of all 51 submitted BioCreAtIvE 2 runs	test	0.6642	0.7636	0.6868	0.7351
Linear SVM, all features	reverse train/test	0.9170	0.8310	0.8720	0.9173
Linear SVM, all features	test crossval	0.7350	0.7520	0.7430	0.8149

References

- [1] Cohen AM, Hersh W. A survey of current work in biomedical text mining. *Briefings in Bioinformatics* 2005;6(1):57-71.
- [2] Cohen AM. Unsupervised gene/protein entity normalization using automatically extracted dictionaries. In: *Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, Proceedings of the BioLINK2005 Workshop*; Detroit, MI: Association for Computational Linguistics. p. 17-24, 2005.
- [3] Tsuruoka Y, Tsujii J. Improving the performance of dictionary-based approaches in protein name recognition. *J Biomed Inform* 2004;37(6):461-70.
- [4] Joachims T. Text categorization with support vector machines: learning with many relevant features. In: *Proceedings of the 10th European Conference on Machine Learning*. p. 137-142, 1998.
- [5] Zhang D, Lee WS. Extracting key-substring-group features for text classification. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. p. 474-483, 2006.
- [6] Cohen AM. An Effective General Purpose Approach for Automated Biomedical Document Classification. In: *Proceedings of the American Medical Informatics Association (AMIA) 2006 Annual Symposium*, 2006.
- [7] Cohen AM, Yang J, Hersh WR. A Comparison of Techniques for Classification and Ad Hoc Retrieval of Biomedical Documents. In: *Proceedings of the Fourteenth Annual Text REtrieval Conference - TREC 2005*; Gaithersburg, MD, 2005.
- [8] Dietterich TG. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation* 1998;10(7):1895-1924.
- [9] Vapnik VN. An overview of statistical learning theory. *IEEE Trans Neural Networks* 1999;10(5):988-999.



A Semi-Supervised Approach To Learning Relevant Protein-Protein Interaction Articles

Mark A. Greenwood

m.greenwood@dcs.shef.ac.uk

Mark Stevenson

m.stevenson@dcs.shef.ac.uk

Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello Street, Sheffield, S1 4DP, UK.

Abstract

This paper describes an Information Extraction system that can be used to identify articles containing protein-protein interactions. The approach relies on the automatic acquisition of dependency tree based patterns which can be used to identify these interactions and consequently select relevant documents. Evaluation shows an F-Score performance of approximately 64%.

Keywords: semi-supervised learning, dependency trees, relation extraction, linked chains

1 Approach

Our approach to the Protein-Protein Interaction (PPI) article subtask (IAS) of the 2nd BioCreAtIvE workshop follows on from previous work on relation extraction which has been applied to several problems including the identification of gene-gene interactions [2]. This method, briefly outlined below, uses a semi-supervised algorithm to learn a relation extraction system given a few example seed patterns which illustrate protein-protein interactions.

Each abstract is pre-processed before extraction patterns are learned. Abstracts are split into sentences. Protein names are identified, using NLProt¹, and substituted with a generic token (PROTEIN). The text is then parsed, using the Stanford parser², to produce a dependency tree for each sentence.

The patterns we use to identify relations consist of chains and linked chains in dependency trees [2]. A chain is a path from a verb to any of its descendants in the dependency tree, passing through zero or more nodes. A linked chain is a pair of chains which share the same verb as their root but do not have any other nodes in common. For example, the linked chain *PROTEIN* $\xrightarrow{\text{subj}}$ *interact* $\xleftarrow{\text{with}}$ *PROTEIN*³ would be found in a dependency parse for the sentence “PROTEIN frequently interacts with PROTEIN”. It has been shown that chain and linked chain patterns are expressive enough to represent the majority of relations within a dependency analysis without generating an unwieldy number of patterns [4].

Space limitations prevent us from describing our learning algorithm in detail, a fuller description is available elsewhere [1]. Briefly, our algorithm for learning linked chain patterns begins with a small number of seed patterns used to provide examples of good patterns. Eight seeds, shown in Table 1, were used for the experiments described here. Our approach extracts all possible chain and linked chain patterns from the corpus and compares each against the seed patterns. Patterns whose similarity score is above a threshold, α , are assumed to be useful extraction patterns and the β of these with the highest score are added to the set of seeds.⁴ This process is then repeated with the remaining patterns being compared against the expanded set of seed patterns. The algorithm continues until no more patterns can be learned.

¹<http://cubic.bioc.columbia.edu/services/nlprot/>

²<http://www-nlp.stanford.edu/software/lex-parser.shtml>

³ $X \xrightarrow{\text{reln}} Y$ indicates that nodes X and Y are connected by the dependency relation *reln* and that X is Y 's daughter.

⁴Based on previous experiments [1], α was set to 0.9 times the score of the best matching pattern and β to 4.

Table 1: Initial Seed Patterns

$PROTEIN \xrightarrow{\text{of}} \text{reduce} \xleftarrow{\text{to}} PROTEIN$	$PROTEIN \xrightarrow{\text{pnmod}} \text{colocalized} \xleftarrow{\text{with}} PROTEIN$
$PROTEIN \xrightarrow{\text{subj}} \text{link} \xleftarrow{\text{obj}} PROTEIN$	$PROTEIN \xrightarrow{\text{subj}} \text{interact} \xleftarrow{\text{with}} PROTEIN$
$PROTEIN \xrightarrow{\text{obj}} \text{connect} \xleftarrow{\text{to}} PROTEIN$	$PROTEIN \xrightarrow{\text{obj}} \text{associate} \xleftarrow{\text{with}} PROTEIN$
$PROTEIN \xrightarrow{\text{subj}} \text{encode} \xleftarrow{\text{obj}} PROTEIN$	$PROTEIN \xrightarrow{\text{subj}} \text{express} \xleftarrow{\text{obj}} PROTEIN$

A key choice in our approach is the method which is used to compare patterns against the seeds. We use a similarity function which is inspired by work on tree kernels [1], although the function used is not itself a kernel. This function compares pairs of patterns by starting at each of their root nodes and comparing their structure until they diverge too far to be considered similar.

Each node n in an extraction pattern has three features associated with it: the word, the relation to a parent, and the part-of-speech (POS) tag. These features are denoted by n_{word} , n_{reln} and n_{pos} respectively. Pairs of nodes can be compared by examining the values of these features and also by determining the semantic similarity of the words. A set of four functions, $F = \{\text{word}, \text{relation}, \text{pos}, \text{semantic}\}$, is used to compare nodes. The first three of these correspond to the node features with the same names and return 1 if the value of the feature is equal for the two nodes and 0 otherwise. The remaining function, semantic , returns a value between 0 and 1 to signify the semantic similarity of lexical items contained in the word feature of each node. This similarity is computed using Lin's lexical similarity function [3] which relies on an information-theoretic measure based on the WordNet hierarchy. The similarity of two nodes, $s(n_1, n_2)$ is 0 if their part of speech tags are different and, otherwise, is simply the sum of the scores provided by the four functions in F .

The similarity of a pair of linked chain patterns, l_1 and l_2 , is determined by the function sim where r_1 and r_2 are the root nodes of patterns l_1 and l_2 and C_r is the set of children of node r . The final part of the similarity function, sim_c , calculates the similarity between the child nodes of n_1 and n_2 . Using this similarity function a pair of identical nodes have a similarity score of four. Consequently, the similarity score for a pair of linked chain patterns can be normalised by dividing the similarity score by 4 times the size (in nodes) of the larger pattern. This results in a similarity function that is not biased towards either small or large patterns but will select the most similar pattern to those already accepted as representative of the domain.

These acquired patterns can then be used to perform the IAS task. The abstracts in the test set are processed, as above, reducing each to a set of patterns. Each abstract is then scored based on the number of acquired patterns it contains. An abstract which does not contain any of the acquired patterns is deemed irrelevant. Relevance of the remaining abstracts is determined by ranking them based on the number of acquired patterns each contains.

2 Results and Analysis

The algorithm ran for 241 iterations before it was unable to acquire any more patterns. Patterns acquired up to iterations 241, 160, and 80 were submitted for the formal evaluation as runs #1, #2 and #3 respectively. After the 80th, 160th and 241st iteration the learning algorithm had acquired 320, 640, and 964 patterns respectively. These were combined with the eight seeds to perform the evaluation task. Results of this evaluation are shown in Table 2. The bottom portion of this table shows the mean and standard deviation, σ , of all 51 submitted runs. These results show that recall increases substantially as the algorithm learns without overly reducing precision, the net result of

Table 2: Official Evaluation Figures

Run	Precision	Recall	Accuracy	F-Score	FPR	TPR	Error Rate	AUC
#1	0.668	0.616	0.655	0.641	0.307	0.616	0.345	0.681
#2	0.735	0.547	0.675	0.627	0.197	0.547	0.325	0.692
#3	0.805	0.373	0.641	0.510	0.091	0.373	0.359	0.664
Mean	0.664	0.764	0.671	0.687	-	-	-	0.735
σ	0.081	0.193	0.064	0.104	-	-	-	0.074

which is an increase in F-Score.

Figure 1 shows the F-Score calculated at each iteration of the learning process. The eight seed patterns achieve an F-Score of 19.6%. The graph demonstrates that there is a steady increase in performance to a maximum of 64.3% (iteration 179), almost 45% more than the seed patterns. The F-Score at the final iteration (64.1%) is slightly lower than the maximum but the graph shows that the algorithm reaches a plateau so that the system submitted as run #1 was close to the best achievable by the system.

Our highest scoring official run and the results from the algorithm’s best performing iteration are comparable with the mean scores of all submitted systems (within one standard deviation). However, our system has the advantage of employing a semi-supervised learning algorithm which requires a very small amount of annotated data (eight seed patterns).

3 Conclusion

This paper has described how an algorithm for learning relation extraction patterns can be used to identify articles containing interactions between proteins. The approach is semi-supervised and requires only a small number of example seed patterns. Analysis shows that the patterns learned by the system improve substantially on the performance of the seeds to produce a system which is comparable to the average score of the systems submitted for this task.

Acknowledgements The research described in this paper was funded by the UK Engineering and Physical Sciences Research Council via the RESuLT project (GR/T06391).

References

- [1] M. A. Greenwood and M. Stevenson. Improving Semi-supervised Acquisition of Relation Extraction Patterns. In *Proceedings of the Information Extraction Beyond The Document Workshop (COLING/ACL 2006)*, Sydney, Australia, 2006.
- [2] M. A. Greenwood, M. Stevenson, Y. Guo, H. Harkema and A. Roberts. Automatically Acquiring a Linguistically Motivated Genic Interaction Extraction System. In *Proceedings of the 4th Learning Language in Logic Workshop (LLL05)*, Bonn, Germany, 2005.
- [3] D. Lin. An Information-theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine learning (ICML-98)*, Madison, Wisconsin, 1998.
- [4] M. Stevenson and M. A. Greenwood. Comparing Information Extraction Pattern Models. In *Proceedings of the Information Extraction Beyond The Document Workshop (COLING/ACL 2006)*, Sydney, Australia, 2006.

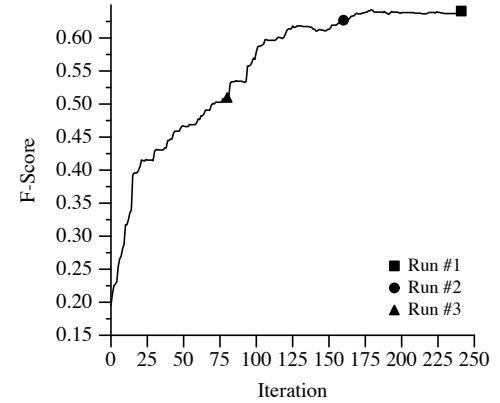


Figure 1: F-Score For All 241 Iterations
■ Run #1
● Run #2
▲ Run #3



ProtIR prototype: abstract relevance for Protein-Protein Interaction in BioCreAtIVe2 Challenge, PPI-IAS subtask

Yan Hua Chen¹
yanhua@idi.ntnu.no

Heri Ramampiaro¹
heri@idi.ntnu.no

Astrid Lægreid²
astrid.lagreid@ntnu.no

Rune Sætre³
satre@is.s.u-tokyo.ac.jp

¹ Department of Computer and Information Science, Norwegian University of Science and Technology, Sem Sælands vei 7-9, NO-7491 Trondheim, Norway

² Norwegian University of Science and Technology, NO-7491 Trondheim, Norway

³ Department of Computer Science, University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033, Japan

Abstract

ProtIR is a prototype developed for the IAS subtask in the BioCreAtIVe2¹ task protein-protein interaction (PPI) extraction. In this paper, we describe the skeleton of the ProtIR system and briefly discuss the results. Our idea is to adapt information retrieval (IR) techniques to solve this task, which is to classify and rank a set of abstracts that may contain a protein interaction. By using a list of well-known protein interaction related keywords, and a list of protein and gene symbols and names collected from the GeneTools' annotation database, we experiment with the bag-of-words approach to explore its advantages and limitations when dealing with biomedical texts. For recognizing a protein mention, we introduced a name evidence scoring scheme, which uses the inverse document frequency (*idf*) as a weighted factor. By including this factor, the system can easier discriminate between terms in the protein name that are specific to the protein and terms that are not. The preliminary result was evaluated by BioCreAtIVe2, and attained an f-score of 68.2% on the test corpus.

Keywords: information retrieval, IR, protein-protein interaction, protein name scoring.

1 Introduction

This paper describes the building blocks of the ProtIR (Protein and Interaction Retrieval) system prototype, its limitations and suggestions for future improvements to this simple baseline system.

The idea is to adapt existing information retrieval (IR) methods [1] to do retrieval and information extraction in the biomedical text domain. The aim of this work is to experiment with this naive approach and assess its baseline performance.

In order to recognize protein-protein interaction (PPI) in an abstract, we used a list of protein/gene symbols and names extracted from the NTNU Annotation Database [2] and a list of protein-protein interaction keywords (more details in [5]). This list is an expansion of the one proposed by Martin et al. [3] that originally consisted of 40 PPI-related words, hereafter referred to as *connection keywords*. In addition, we apply a scoring scheme that favors proteins that are mentioned together with their full names.

The system consists of seven modules, as shown in figure 1. The main modules M1-M5 are responsible for tokenization, indexing, term categorization, evidence score calculation and relevance classification, respectively. MA and MB are support modules for term categorization and evidence score calculation. The former is an independent index of reference protein symbols with their synonyms and full names, and the latter deals with weight calculation for the terms in each of the protein names. The modules are implemented

¹ BioCreAtIVe2 challenge for text mining and IE (http://biocreative.sourceforge.net/biocreative_2.html).

in a prototype using Java 1.5, Lucene 2.0 and MySQL 5.0. Supplementary information is to be found in [5].

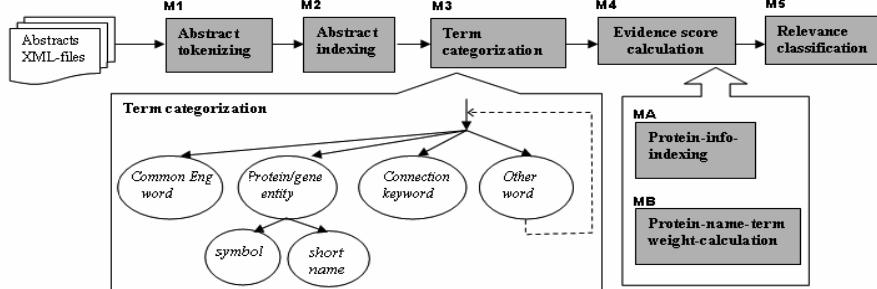


Figure 1: System overview of ProtIR, including modules M1-M5 and support modules MA and MB.

2 Method

As can be inferred from the above figure, our approach is pipeline based, starting with pre-processing, parsing and indexing of the abstracts. First, in module M1, the raw abstract text is parsed and tokenized by a specially designed analyzer that removes meaningless special characters. Second, an index is constructed from the tokenized abstracts in module M2. This index provides a list of term-frequency vectors that are convenient to use in the next modules for protein and interaction recognition.

Further, the evidence of PPI is extracted in two steps. The first step, term categorization, is to find the relevant protein/gene candidates and the interaction mention. The second step is to score these proteins by scanning for the evidence of their full names.

2.1 Term Categorization (M3)

As shown in Figure 1, the terms from the abstract index are categorized into four categories: common words, protein/gene entities (e.g. names or symbols), connection keywords and other words.

To recognize a protein/gene entity, we use a list of symbols and short protein names, i.e. single term names. This list is constructed by extracting relevant fields from the annotation database. It is filtered for ambiguous symbols and names, i.e. common English words and ordinary abbreviations. For this filtering, we use a list of common words from sourceforge [4] with some modifications, and a list² of nouns and stop words that are common in protein names. In addition, improper names are filtered out using simple regular expression rules. The candidate interactions are recognized by using the list of connection keywords in [5]. These are the words that are often related to protein interactions. The last category contains unrecognized words. These may include protein symbols, protein names and connection keywords joined with another word or symbol.

2.2 Evidence Score Calculation (M4)

In our scoring scheme, the relevance calculation for PPI is based on the occurrences of protein entities and connection keywords. The terms categorized from the previous module are scored by their term-frequency multiplied with a *term score*.

For a connection keyword, the score is set to be a constant. For a protein candidate, the score is calculated using the support modules MA and MB to check for the evidence of the corresponding protein/gene name in the text. As shown in the protein scoring algorithm in Figure 2, the protein score for a short protein name is a constant, whereas for a symbol, it is a sum of the weights of all the name terms that are specific to the reference protein symbol. Finally, boosting values were chosen for each of the following cases: the protein occurs in the title, the protein is bound to a connection keyword in the text, and the protein symbol is the same as its name.

The MA module manages a separate index that handles the correlation between an official protein symbol, its full name, aliases and synonyms from SwissProt, UniGene and Entrez Gene extracted and collected from the Annotation Database [2]. The MB module is responsible for the name term weighting scheme. It contains

² A list of nouns and stop words that are common in protein names was built from the GENIA corpus.

a database that restructures the information in the MA index. The database holds the relationships between the protein/gene symbols, the terms in their full names and the weight of each of these terms. For a term in the corresponding full name of a protein symbol, the weight is a product of the term frequency (tf) in the name and the inverse document frequency (idf) of that term for all protein names in MA. Idf is used here to normalize the weights of common terms and to emphasize terms that are specific to a protein name. This is a way to verify a protein symbol and disambiguate it from other non-protein symbols and abbreviations. Due to lack of space, we refer to [1] for the definition of tf and idf .

1. for all protein candidates p in the document d do
2. if p is a short protein name, then
 3. $protein\ score = constant\ C$
 - else (if p is a symbol)
 4. for all official symbols s related to p (p may be a synonym or alias) do
 5. for all terms t in the protein full name for the symbol s do
 6. if term t exist in the document d , then
 7. $protein\ score +=$ the weight of the term t in the protein-term-weight-DB

Figure 2: Protein scoring algorithm.

2.3 Relevance Classification (M5)

As can be inferred from the previous section, PPI is detected by co-occurrence of protein and connection keyword. Its relevance is calculated as the sum of the evidence scores for each term indicating a protein or an interaction in the abstract. To classify whether an abstract contains relevant PPI information, two score thresholds were determined by the level of precision and recall from the prediction of the training data.

3 Results and Discussions

The prototype was trained on a sub collection of 90% of the abstracts from the true positive and true negative sets from BioCreAtIVE2. The rest of the collection was used for preliminary testing. This test attained an f-score of 77.7%, while for the BioCreAtIVE2 test collection the system only attained 68.2%. This difference reflects the common sampling issue between training and test data. Aside from that, it may also reflect that the system is not able to recognize all protein symbols, which may be caused by the incompleteness of the list of protein/gene symbols and names extracted from the Annotation Database of NTNU. One way to solve this problem is to integrate more information to this list. In addition, applying other existing named entity recognition (NER) techniques and tools seems inevitable to improve the protein entity recognition phase.

For further improvements, we also consider to focus on sentence-based recognition of protein names and protein interaction relationships. Moreover, a module for monitoring and verification of the relationships between the protein names, symbols and synonyms extracted from the text is a possible extension for future work. Machine learning may also be applied to adjust the boosting values in the relevance scoring scheme.

References

- [1] Baeza-Yates R. and Ribeiro-Neto B., *Modern Information Retrieval*, ACM Press, 1999
- [2] Beisvag V., Jünge F. K., Bergum H., et al., GeneTools – application for functional annotation and statistical hypothesis testing. *BMC Bioinformatics*, 7(470), 2006.
- [3] Martin E. P. G., Bremer E. G., Guerin M.-C., et al. Analysis of Protein/Protein Interactions Through Biomedical Literature: Text Mining of Abstracts vs. Text Mining of Full Text Articles. *Proc. The Knowledge Exploration in Life Science Informatics (KELSI)*. Springer-Verlag Heidelberg.96-108. 2004.
- [4] <http://wordlist.sourceforge.net/>
- [5] <http://www.idi.ntnu.no/~yanhua/biocreative/>



A Term Investigation and Majority Voting for Protein Interaction Article Sub-task 1 (IAS)*

Man LAN¹

lanman@comp.nus.edu.sg

Chew Lim TAN¹

tancl@comp.nus.edu.sg

Jian SU²

sujian@i2r.a-star.edu.sg

¹ School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543

² Institute for Infocomm Research, 21 Hen Mui Keng Terrace, Singapore 119613

Abstract

The BioCreAtIvE II PPI Interaction Article Sub-task 1 (IAS) is a biological text classification task which concerns whether a given abstract contains protein interaction information. In order to improve the performance of text classification, we examined ways to represent text from the term type and term weighting aspects. In addition, we also combined different classifiers by majority voting technique.

Keywords: biological text classification, text representation, named entity, term weighting

1 Introduction

For general text classification task, vector space model is usually adopted to represent the text. Thus, there are two issues of text representation involved, i.e. (1) what should a term be and (2) how to weight a term. In this work, we investigated different text representations for biological text classification from the above two aspects. That is, we adopted a protein name-based representation and a new effective term weighting method based on our two previous studies in [1] and [2]. Based on our knowledge, so far no such work has been done on biological text classification from the two representation aspects. Moreover, we also explored several machine learning algorithms to build the classifier.

2 Methodology and Results

2.1 Text Preprocessing

The BioCreAtIvE II PPI IAS training corpus consists of 3536 positive and 1959 negative documents on which this constructed system is based on. The Porter's stemming was performed to reduce words to their base forms. Stop words (513 stop words), punctuations and numbers were removed. The threshold of the minimal term length is 3 (many biological keywords contain 3 letters, such as acronym). The resulting vocabulary has 24648 words (terms or features). By using the χ^2 statistics ranking metric for feature selection, the top $p = \{200, 300, 400, 450, 500, 1000, 1500\}$ features per positive and negative category were selected from the training set.

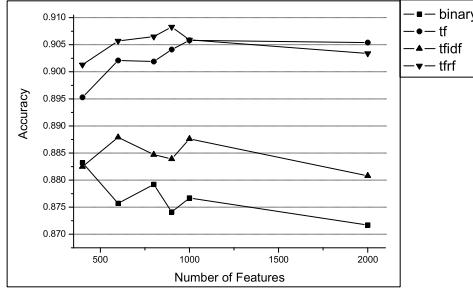
2.2 Preliminary Results on the Training Corpus

2.2.1 Performance of Different Term Weighting Methods for Text Classification

Different features have different importance in a text and thus an important indicator represents how much the feature contributes to the semantics of document. Our proposed weighting method, i.e. *tf.rf* [2], is based on the idea that the more concentrated a high frequency term is in the positive category than in the negative category, the more contributions it makes in selecting the positive documents from among the negative documents. We name it *rf* (relevance frequency) because only the frequency of relevant documents (i.e. those which contain this term) are considered and it is calculated as the ratio of relevant documents in the positive and negative category (in stead of using whole document distribution in the corpus). It has shown consistently better performance than other traditional methods based on cross-method, cross-classifier and cross-corpus validation. In this work, we chose four methods, i.e. *binary*,

*The work is partially supported by a Specific Targeted Research Project(STREP) of the European Union's 6th Framework Programme within IST call 4, Bootstrapping Of Ontologies and Terminologies STrategic Project(BOOTStrep).

binary. *tf* has shown good performance even though sometimes it had a bit lower accuracy than *tf.rf*. On the other hand, the widely-used *tf.idf* method only performed better than *binary* method. These findings are consistent with those in our previous work [2]. Meanwhile, since the best performance has been achieved using *tf.rf* with 900 features (bag-of-words), we chose these settings in the following test experiment.



Scheme	Micro- <i>P</i>	Micro- <i>R</i>	Micro- <i>F</i> ₁
<i>binary</i>	92.55 ± 0.94	91.99 ± 4.05	92.22 ± 1.77
<i>tf</i>	92.34 ± 1.23	94.26 ± 3.26	93.17 ± 1.43
<i>tf.idf</i>	92.19 ± 1.01	94.48 ± 3.69	93.28 ± 1.64
<i>tf.rf</i>	92.23 ± 1.24	95.11 ± 2.79	93.63 ± 1.23

Table 1: Best results of four term weighting schemes.

Figure 1: Results of four term weighting methods.

2.2.2 Performance of Named Entity-based Representation for Text Classification

The protein entity names in biology domain are more complex than those in other domains like newswire due to the long descriptive naming convention, non-standardized naming convention and ambiguous abbreviation, etc. Based on the consideration that protein named entity may capture more information left out of the bag-of-words approach, we conducted experiment using this representation on this corpus. We adopted an existing named entity recognition system named PowerBioNE [1], where various evidential features are integrated through a Mixture Markov Model(HMM)-based named entity recognizer.

The noticing phenomena of these extracted named entities are sparse and skewed distribution. First, most named entities are in the positive documents (76.7%) and only few (23.3%) are in the negative documents. This is reasonable since the positive documents are relevant to protein interaction articles and thus they must contain more protein names than negative documents. Second, most of the named entities occur only once or few times in the corpus. For example, 25740 named entities (83.7%) occur only once, 2529 entities (8.2%) occur more than three times and only 380 entities (1.2%) occur more than ten times in the whole corpus. This sparse distribution problem make the document indexing difficult since many documents will be represented as null vectors when the number of named entities used for indexing is quite small. Therefore, we also combined named entity-based representation with the bag-of-words approach. Table 2 shows the results of these combined different representations, where NE denotes

Scheme	Micro-P	Micro-R	Micro-F1
NE(<i>tf</i>)	68.03 ± 0.81	92.98 ± 2.76	78.56 ± 1.28
NE+BOW(<i>binary</i>)	91.51 ± 0.94	92.98 ± 4.05	92.20 ± 1.77
NE+BOW(<i>tf</i>)	91.90 ± 1.19	94.74 ± 2.76	93.27 ± 1.35
NE+BOW(<i>tf.rf</i>)	91.97 ± 1.19	95.16 ± 2.76	93.52 ± 1.35

Table 2: Results of different combined represents on the BioCreAtIVe II corpus.

Classifier	Accuracy
LibSVM	0.9083 ± 0.0011
kNN	0.7821 ± 0.0013
AdaBoost	0.8667 ± 0.0094
Voted Perception	0.8917 ± 0.0080
Majority Voting	0.9099 ± 0.0023

Table 3: The results of classifier committee.

named entity and BOW means bag-of-words approach. Based on the results from Table 1 and Table 2,

we can find that named entity-based representation was the most disappointing. It only achieved 78.56% F_1 score. When combined with the bag-of-words approach based on different term weighting methods, the named entity-based representation has not increased the performance of text classification.

2.2.3 Performance of Different Classifiers

Generally, SVM has been confirmed to perform better than many promising machine learning algorithms. In addition, since different high-quality classifiers make at least practically uncorrelated errors, and when combined with a majority voting, they (i.e. classifier committee) are expected to lead to higher performance. We also explored majority voting technique in this work. Table 3 lists the performance (accuracy) using different algorithms with 900 features and *tf.rf* scheme based on two-folder cross validation.

2.3 Results on the Test Data and Error Analysis

According to the above experimental results and system settings on the training data, Table 4 lists the three sets of system configuration and the corresponding different evaluation scores on the 750 test documents (350 positive and 350 negative documents), where AUC means the area under the ROC curve.

Run	#_features	weighting	Classifier(s)	Accuracy	F_1	AUC
1	900(BOW)	<i>tf.rf</i>	LibSVM	0.7467	0.7775	0.8141
2	900(BOW)	<i>tf.rf</i>	Classifier Committee	0.7453	0.7761	0.8105
3	800(BOW)	<i>tf.rf</i>	Classifier Committee	0.7373	0.7685	0.8019

Table 4: The system configuration and results on the test data.

To further evaluate our system and explore possible improvement, we have implemented an error analysis. The average error rate of our system is 0.2569. First, the reason for protein name-based representation failing to improve the performance may be caused by the precision of named entities extracted. Although PowerBioNE achieved 77.8% F_1 score on the “protein” class of GENIA V3.0 which is higher than other systems [1], the accuracy of extracted named entities is still not high. The system performance can be improved further by using more annotated corpora and incorporating more effective features based on the domain knowledge. On the other hand, the accuracy performance on the training corpus using *tf.rf* is above 90% while it only achieved 74% on the test data. The possible reason may be caused by the different category distribution in the training and test data, i.e. the ratio of positive and negative documents in the training corpus is almost 2:1 while it is 1:1 in the test corpus. The term weighting is calculated based on the distribution of the training corpus and used for the test corpus. This may cause the significantly different performance in the training and test corpora.

3 Concluding Remarks

Our proposed *tf.rf* method showed classification power in biological text classification while named entity-based representation has not yet succeeded in improving text classification performance over the bag-of-words approach. We should point out that the observations above are made based on the controlled experiments and the accuracy of extracted named entities also has an effect on the result. We believe more advanced NLP techniques and advanced ways of incorporating NLP output could further improve the performance of text classification, for example, high performance coreference resolution to normalize the protein names through different variations, nominal or pronominal expressions could generate more occurrences of the same protein names to facilitate the further text classification.

References

- [1] GuoDong Zhou and Jian Su. Exploring deep knowledge resources in biomedical name recognition. *Proceedings of JNLPBA shared task*, 99–102, 2004.
- [2] Man Lan, ChewLim Tan and HweeBoon Low. Proposing a New Term Weighting Scheme for Text Categorization. *Proceedings of the 21st AAAI*, 763–768, 2006.



Identifying Protein-Protein Interaction Sentences Using Boosting and Kernel Methods

Soo-Yong Shin¹³

syshin@nist.gov

Sun Kim²³

skim@bi.snu.ac.kr

Jae-Hong Eom²

jheom@bi.snu.ac.kr

Byoung-Tak Zhang²

btzhang@bi.snu.ac.kr

Ram Sriram¹

sriram@nist.gov

¹ Manufacturing Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA

² Biointelligence Laboratory, School of Computer Science and Engineering, Seoul National University, Seoul 151-744, Korea

³ Both authors have equally contributed to this work.

Abstract

As the amount of biological research literature increases, finding information is becoming a daunting task. Since machine learning techniques could alleviate this problem, we propose a machine learning framework to identify protein-protein interaction sentences from research papers. This machine learning technique is one of the basic components needed to automatically extract biological information from texts. Since the protein-protein interaction (PPI) sentences have their own patterns at article and sentence levels, these patterns are mined by using boosting and kernel methods. Both approaches have good characteristics for the PPI extraction tasks, and naturally can handle heuristic information for future extensions.

Keywords: Protein-Protein Interaction Identification, Boosting Methods, Tree Kernels, Support Vector Machines

1 Introduction

The growing accumulation of functional descriptions in biomedical literature necessitate the use of text mining tools to facilitate the extraction of such information [1]. Therefore, diverse approaches such as pattern matching, statistical learning, and natural language processing have been proposed. Here, we present a machine learning-based framework, in particular, without any prior knowledge other than training data. In biological text mining, only a small amount of annotated documents are available for public use, which limits the usage of machine learning techniques. Nevertheless, it is important to examine the ability of machine learning methods to determine the possibility for real-world use, because the heuristic approaches (with or without learning) need too much efforts of human experts.

The goal of the BioCreative project is to evaluate text mining and information extraction systems applied to the biological domain [1]. We participated in two subtasks of the Protein-Protein Interaction (PPI) task in the BioCreative II competition. The subtasks of PPI of interest to us are the Protein Interaction Article (IAS) subtask and the Protein Interaction Sentence (ISS) subtask. The IAS subtask is the classification of whether a given article contains protein interaction information. It is the first step to extract the PPI information, by selecting those articles which have relevant information related to protein interactions. The IAS system should return a ranked list of PPI articles based on their relevance in the task. Before getting protein interaction pairs, it is useful to select the most relevant sentences which are directly connected to the protein interactions. The ISS subtask is to filter those PPI relevant sentences. The ISS system is required to submit a ranked list of HTML passages describing protein-protein interactions.

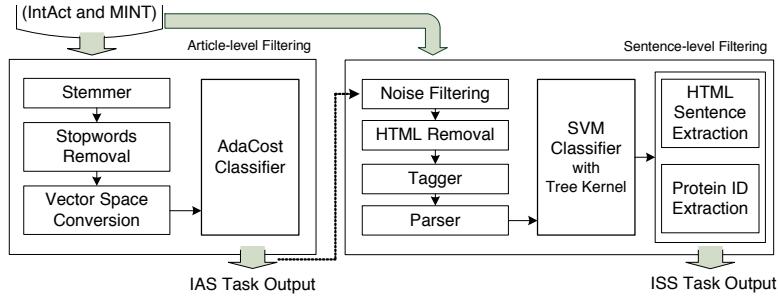


Figure 1: Overview of the PPI extraction system for BioCreative II. Dotted line is not implemented and not used for the BioCreative tasks as the committee provides two separate training sets.

For the IAS subtask, the AdaCost [5], a cost-sensitive learning algorithm, is used to give bias towards PPI relevant documents. Since we use naive Bayes classifiers as weak learners in the AdaCost framework, any prior knowledge can be naturally adapted in probabilistic form. For the ISS subtask, a tree kernel method [4] is utilized to mine the PPI patterns among sentences, which is based on the assumption that the PPI information tends to be written in specific grammatical structure [6]. It also can employ additional heuristic knowledge in an easy way.

The paper is organized as follows: In Section 2, the proposed PPI extraction approaches are described and analyzed. Concluding remarks and future research are provided in Section 3.

2 Methods and Analysis

The proposed PPI extraction system consists of two parts: 1) article-level and 2) sentence-level filters. These filters are for the IAS subtask and the ISS subtask, respectively. Figure 1 shows the overview of the two-phase PPI extraction. Free texts enter the article-level filter at first, which identifies the PPI relevant articles using the AdaCost classifier. After the PPI articles are classified, the PPI information is picked up at the sentence level. The second phase uses support vector machines (SVMs) with tree kernels. Although the article-level and the sentence-level filters are combined together as a complete PPI extraction system, the two phases are separately performed for the BioCreative tasks, and each produces its own result according to the participating subtasks.

2.1 PPI Article Filtering by Cost-Sensitive Learning

The IAS subtask is the first step to extract the PPI information at article level, so that the actual extractor (ISS system) can use less-noisy data. At this point, the filtering system should not miss any PPI relevant document even though a certain amount of irrelevant documents are included in the filtered set, i.e., recall is more important than precision. To handle the tradeoff between recall and precision, our system utilizes a cost-sensitive learning algorithm, AdaCost [5]. Unlike other machine learning classifiers, which focus on minimizing the number of incorrect predictions, AdaCost provides the flexibility between precision and recall rates by using a cost factor. It is similar to AdaBoost [8], but the main difference is how the data distribution is updated. AdaCost has an additional parameter, so-called “cost” in updating the data distribution. The weight of an instance with high cost will be changed more than the weight of an instance with low cost. This allows the learning system to classify high-cost instances more correctly. We use naive Bayes learning as a weak learner which is known to be efficient in text filtering [7]. In addition, the naive Bayes classifier is suitable for our purpose of

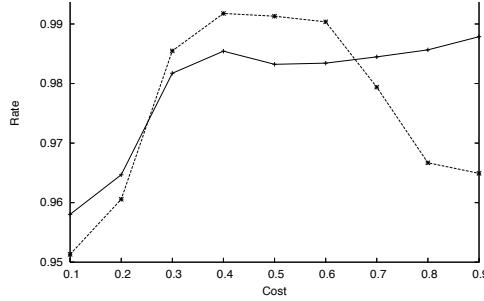


Figure 2: Recall and F-score changes for cost on unbalanced dataset.

participating in the BioCreative II, which is to build a machine learning framework that can be further used to adapt heuristic knowledge in easy ways. The naive Bayes classifier is a statistical learning method that can naturally use the heuristic knowledge only if it can be transformed into probabilities. The modified AdaCost with naive Bayes algorithm used for the article-level filtering is as follows (our modification is shown in bold letters):

- Given training examples $S = \{(x_1, c_1, y_1), \dots, (x_m, c_m, y_m)\}$; x_i is an instance ($x_i \in X$), c_i is a cost factor ($c_i \in R^+$), and y_i is a label ($y_i \in \{-1, +1\}$).
- Initialize $D_1(i)$ (such as $D_1(i) = c_i / \sum_j^m c_j$).
- For $t = 1, \dots, T$:
 1. **Train a naive Bayes classifier using distribution D_t .**
 2. Compute weak hypothesis $h_t : X \rightarrow R$.
 3. Choose $\alpha_t \in R$ and $\beta(i) \in R^+$, where α_t is a weight parameter for weak hypothesis h_t at the t -th round, and $\beta(i) = \beta(\text{sign}(y_i h_t(x_i)), c_i)$ is a cost-adjustment function.
 4. Update $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i) \beta(i))}{Z_t}$, where Z_t is a normalization factor.
- Output the final hypothesis:

$$H(x) = \text{sign}(f(x)) \text{ where } f(x) = \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

The training data used in the IAS subtask contains 3,536 positive examples and 1,959 negative examples. The noisy positive examples given by the committee are excluded from the experiments. For the AdaCost classifier, the documents are preprocessed by stemming and stopword removal [7]. We use a modified stopword list, where the PPI-related words are omitted from common stopwords. Then the remaining texts are converted to the bag-of-words representation because we presume that some specific words or the simple combination of the words are enough to evaluate the PPI relevance of the articles.

Figure 2 presents recall and F-score changes for the cost c_i on training data. The overall best performance occurs at 0.4 cost, whereas the highest recall is achieved at 0.9 cost. The unusual peak of 0.4 cost is caused by the unbalanced number of positive and negative examples and relatively small size of dataset. In the article-level filtering, the recall is more important unless the F-score drops drastically, hence higher cost is preferred. However, for the official run of the IAS subtask, the cost was set to 0.5 since it is an independent subtask from other PPI subtasks, and only evaluated by the IAS system output. Our IAS system got 65.73 % of accuracy and 71.54 % of F-score on test data.

We found out that there is a different PPI-related vocabulary between training examples and test examples, which bears the performance decrease on test data. This problem can be solved by using PPI-related dictionaries or databases, which remains as future research.

2.2 PPI Sentence Filtering by Tree Kernels

The ISS subtask consists of two steps: choosing relevant sentences and finding UniProt IDs of interacting protein pair. For the first step, we assume the PPI sentences can be discriminated by investigating their grammatical structures, since most of PPI sentences tend to have unique grammatical structures [6]. A parsing tree in natural language processing represents a set of words and its structural information. The convolution kernel was chosen to calculate structural similarity among parsing trees [4].

In the convolution tree kernel algorithm, kernel value is evaluated by summing up the number of common subtrees between two trees to calculate the structural similarity. A tree is represented as a vector of subtrees through high dimensional feature mapping [4]:

$$\Phi(\text{Tree } T) = (\text{subTree}(\text{type 1}), \dots, \text{subTree}(\text{type } n)),$$

where $\text{subTree}(\text{type } n)$ is the number of subtree of node type n . Then, the kernel function is defined as follows:

$$K(T_1, T_2) = \langle \Phi(T_1) \cdot \Phi(T_2) \rangle = \sum_l \Phi(T_1)[i] \times \Phi(T_2)[i] = \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \sum_i I_i(n_1) \times I_i(n_2),$$

where N_1 and N_2 represent the set of all possible nodes of trees T_1 and T_2 , and $I_i(n)$ is an indicator function which has 1 if sub-tree of type i starts from root node n , 0 otherwise.

The number of subtrees with type i in tree T is calculated by $\Phi(T)[i] = \sum_{n \in N} I_i(n)$, which gives the total number of nodes in tree T which have subtrees with type i . The inner product between two trees, having its features as the all possible subtrees, is computed by the following recursive way and it is known to be calculated in polynomial time.

- If the form of the child nodes of n_1 and n_2 are different, $NCS(n_1, n_2) = 0$, where $NCS(n_1, n_2)$ is the number of common subtree between n_1 and n_2 .
- If the form of the child nodes of n_1 and n_2 are identical (including their order) and they are leaf nodes, $NCS(n_1, n_2) = \lambda$.
- For all other cases, $NCS(n_1, n_2) = \prod_j (1 + NCS(ch(n_1)_j, ch(n_2)_j))$, where $ch(n_1)_j$ is the j -th child of node n_1 , $ch(n_2)_j$ is the j -th child of node n_2 , and $NCS(ch(n_1)_j, ch(n_2)_j) = \lambda \sum_i I_i(n_1) \times I_i(n_2)$. The parameter λ , $0 < \lambda \leq 1$, is used to consider the relative importance of tree fragment according to its length and is set to '1' when the size of tree fragments is not considered.

To achieve the parsing tree of the sentence, we use the following procedure. First, we extract plain texts by removing HTML tags in HTML documents to use the grammatical structure information. Second, the extracted sentences are tagged by a rule-based part-of-speech tagger [2]. The Brill tagger is trained beforehand, using GENIA corpus (available at <http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus>). Third, the tagged sentences are parsed by a statistical natural language parser [3]. Then, the irrelevant parsing trees such as a noun phrase are discarded since they do not contain the meaningful grammatical structure. This leads to some positive examples that only have noun phrases be excluded from training data. After calculating the tree kernel, the interaction patterns are learned by support vector machines (SVM). We use the LIBSVM package (available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>) which can handle pre-computed kernel matrices.

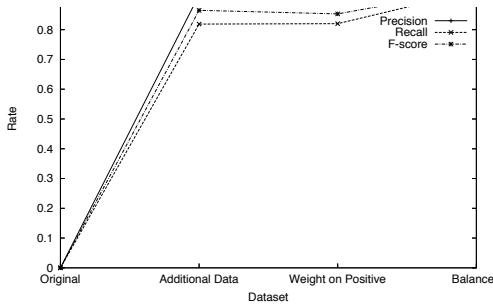


Figure 3: Precision, recall and F-score changes for training data sets.

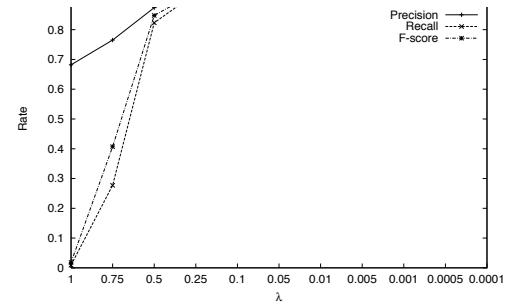


Figure 4: The effect of λ in tree kernel calculated with balanced dataset.

We try to balance the positive and negative examples to improve the quality because the excessive negative examples in the original dataset force the SVM classifier to turn all test sentences into negative examples. We incorporate the Anne-Lise-Veuthey corpus and the PRODISEN Interaction corpus to enrich the positive examples, which are also released by the committee for the ISS subtask. And we also choose the part of the original dataset to reduce the size of negative examples. Finally, Training data for the ISS subtask consists of 1,634 positive sentences and 1,763 negative sentences. For the official run, we use about 10 % more negative examples than positive ones, so that we give a slight bias to non-relevant PPIs, and can get reduced computational time. Note that only a few sentences are available as positive examples at sentence-level filtering out of whole texts.

Figure 3 shows the performance changes for 4 different training data sets. The results were obtained from 10-fold cross-validation. The “Original” means the first standard dataset provided by the ISS subtask. The “Additional Data” is created by adding the corpus, Anne-Lise-Veuthey and PRODISEN Interaction, and the second standard dataset to the “Original.” The “Weights on Positive” gives more weight to positive examples in the “Additional Data.” The “Balanced” is the balanced dataset, where the number of negative examples is only 10 % more than that of positive examples. The balanced dataset gains the best performance, and it shows the importance of making balances between positive and negative examples. The effect of λ in the tree kernels was also examined. Figure 4 shows the experimental results. Since sentence lengths are very diverse, λ should be carefully chosen. The best performance is taken when λ is 0.01, and we got 94.30 % precision, 93.15 % recall, and 93.72 % F-score on the balanced training data. According to the preliminary results, we found that the tree kernel provides good predictions if the corpus is limited to certain conditions for both training and test data.

In the submitted run of the ISS subtask, we used the reduced sentences which removed the words tagged by less important elements such as articles, adverbs, and adjectives to save computational time. However, the follow-up experiments showed that using original sentences provides better performance for all criteria. Because the answers for the ISS test data have not been published yet, we could not analyze the proposed method and its variants further.

Even though we concentrated on the HTML sentence extraction, we also implemented the protein ID extraction module using a simple word-to-word matching approach to find protein IDs from the selected PPI sentences. A UniProt ID dictionary is built with gene names, aliases, orf names, and protein descriptions. Simple morphological variations for each protein term are considered to increase the coverage in the searching process. We also consider compound words by using bi-gram and tri-gram of a sentence. Finally, the nearest two UniProt IDs found in a sentence are selected as a system result.

3 Summary

We presented a machine learning approach to extract protein-protein interactions. This method consists of two procedures: article-level filtering and sentence-level filtering. In the article-level filtering, documents are roughly classified to reduce the overhead in the second procedure. The AdaCost with naive Bayes classifiers is used for the article-level filtering, and the SVM classifier with tree kernels is used to identify PPI relevant sentences as sentence-level filtering.

Our focus is to develop a machine learning-based framework, which can be further enhanced by adding heuristic techniques because it extends the system performance particularly in the biomedical domain. In the present work, we did not apply any heuristic approaches such as protein/interaction word dictionaries. Previous research indicates that the dictionary method could increase the PPI extraction performance when the training data size is limited. Thus, study on exploring efficient heuristic approaches remains as a future research work.

Acknowledgments

This work was supported by the Korea Science and Engineering Foundation(KOSEF) through the National Research Lab. Program funded by the Ministry of Science and Technology (No. M10400000349-06J0000-34910). Soo-Yong Shin was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-214-D00140) and the Manufacturing Metrology and Standards for the Health Care Enterprise Program at NIST. Jae-Hong Eom was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2006-511-D00355).

Mention of commercial products or services in this paper does not imply approval or endorsement by NIST, nor does it imply that such products or services are necessarily the best available for the purpose.

References

- [1] C. Blaschke, E. A. Leon, M. Krallinger, and A. Valencia, Evaluation of BioCreAtIVe Assessment of Task 2, *BMC Bioinformatics*, 6(Suppl 1):S16, 2005.
- [2] Brill, E., A Simple Rule-Based Part-Of-Speech Tagger, *Proc. 3rd Conf. on Applied Natural Language Processing*, 152–155, 1992.
- [3] Collins, M., Head-Driven Statistical Models for Natural Language Parsing, *PhD Dissertation*, University of Pennsylvania, 1999.
- [4] Collins, M. and Duffy, N., Convolution Kernels for Natural Languages, *Proc. 15th Conf. on Neural Information Processing Systems*, 625–632, 2001.
- [5] Fan, W., Stolfo, S., Zhang, J., and Chan, P., AdaCost: Misclassification Cost-Sensitive Boosting, *Proc. 16th Inter. Conf. on Machine Learning*, 97–105, 1999.
- [6] Jang, H., Lim, J., Lim, J.-H., Park, S.-J., Lee, K.-C., and Park, S.-H., Finding the Evidence for Protein-Protein Interactions from PubMed Abstracts, *Bioinformatics*, 22(14):e220–e226, 2006.
- [7] Kim, Y.-H., Hahn, S.-Y., and Zhang, B.-T., Text Filtering by Boosting Naive Bayes Classifiers, *Proc. 23rd Inter. ACM SIGIR Conf.*, 168–175, 2000.
- [8] Schapire, R. E. and Singer, Y., Improved Boosting Algorithms Using Confidence-rated Predictions, *Machine Learning*, 37(3):297–336, 1999.



OntoGene in Biocreative II

Fabio Rinaldi¹ Thomas Kappeler¹ Kaarel Kaljurand¹
Gerold Schneider¹ Manfred Klenner¹ Michael Hess¹
Jean-Marc von Allmen² Martin Romacker² Therese Vachon²

¹ Institute of Computational Linguistics, University of Zurich,
Binzmühlestrasse 14, CH-8050 Zurich , Switzerland

{rinaldi,gschneid,klenner,kalju,hess}@ifi.unizh.ch

² Novartis Pharma AG, Basel, Switzerland,
{martin.romacker,jean-marc.von.allmen,therese.vachon}@novartis.com

Abstract

Research scientists and companies working in the domains of biomedicine and genomics are increasingly faced with the problem of efficiently locating, in the vast amount of published scientific results, the critical pieces of information that are needed in order to assess current and future research investment.

In this paper we describe approaches taken within the scope of the second Biocreative competition in order to solve two aspects of this problem: the detection of novel protein interactions reported in scientific articles, and the detection of the experimental method that was used to confirm the interaction.

Our approach is based on a high-recall protein annotation step, followed by two sharp disambiguation steps. The remaining proteins are then combined according to a number of lexico-syntactic filters, which deliver high-precision results, while maintaining a reasonable recall.

1 Introduction

The increasing amount of published scientific results in the domains of biomedicine and genomics poses, to research scientists and companies alike, the problem of efficiently locating the most relevant pieces of information. The research community is therefore keen to adopt novel Text Mining solutions, which have the potential of supporting such discovery process [3]. While there is a broad consensus on the need for Text Mining, there is still a lot of controversy on which of the many possible approaches are most suited for each specific goal.

In this paper we describe experiments performed within the scope of the most recent BioCreAtIVe¹ competition, using tools developed within the scope of the ONTOGENE project.² BioCreAtIVe is ideally suited to create the conditions necessary for significant scientific advance in the area of Text Mining.

The ONTOGENE project aims at developing and refining (semi-)automatic methods for the discovery of interactions between biological entities from the scientific literature. The ONTOGENE approach is based on dependency-based linguistic analysis of scientific articles [6]. As witnessed by a number of recent publications [1, 2, 4], there is a growing interest in dependency-based representations for the purpose of biomedical Text Mining. One of the advantages of dependency based syntactic representations is that they can be mapped easily into a semantic representation, or, by application of simple transformations, can be used directly to match candidate answers with given queries, allowing easy identification of the arguments of complex relations [5].

In the rest of this paper we describe first the approach followed for subtask 3.2 (IPS). More specifically, section 2 presents our approach to detection of proteins in text, their annotation, and the various disambiguation steps that we have followed. In section 3 we describe how the possible interactions among proteins are generated and selected. Finally, section 4 describes the approach adopted for subtask 3.4 (IMS).

2 Identification and selection of Interactors

It is well known that protein names are highly ambiguous. Researchers working in specific sub-communities tend to develop their own nomenclature, resulting in multiple names for the same protein. Acronyms and abbreviations further complicate the picture. Simply being able to recognize a protein name as such is just a

¹<http://biocreative.sourceforge.net/>

²<http://www.ontogene.org/>

starting point. The name needs then to be unambiguously qualified, by referring it to an entry into a standard protein database, such as UniProt.³

In order for that to happen, disambiguation must happen at two levels: interspecies (i.e. to which specific organisms does the protein mention refer) and intraspecies (i.e. within a given organism, which specific protein is meant). For example, a protein mentioned in text as **eIF4E** could refer to a large number of different proteins. A search in the SwissProt section of UniProt (the manually curated section), delivers 46 possible matches. However if the term appears contextually with the mention of a specific organism, like in the sentence "*The Cap-binding protein eIF4E promotes folding of a functional domain of yeast translation initiation factor eIF4G1*", then it is reasonable to assume that the name refers to a specific organism (yeast), thus restricting the possible matches in UniProt to the following two: EAP1_YEAST (**eIF4E-associated protein 1**) and IF4E_YEAST (**Eukaryotic translation initiation factor 4E**). For the task of protein annotation we have adopted a high-recall low-precision term annotation approach, followed by very strict disambiguation steps, which gradually increase precision (at some expense for recall).

UniProt contains for each protein a list of frequently used synonyms. We have built a database which maps the synonyms to the protein identifier. We have further enriched such list using morpho-syntactic rules that generate variants of the synonyms. Starting from a version of UniProt which contained 228670 protein identifiers⁴, we extracted a list of 203061 unique protein names, and, after generation of the variants, obtained a DB of 698365 terms. Those terms are by necessity highly ambiguous: in average each term refers to 3 proteins, but there are also some terms referring to hundreds of proteins.

Because of the far from perfect HTML-to-text conversion of the articles, we decided early on to use only the abstracts, which we automatically downloaded, in plain text format, from PubMed.⁵ We work on the assumption that the authors will mention in the abstract the most relevant interactions that they discover (although in some cases this might not be true). The input abstracts are tokenized using a custom tokenizer. The stream of tokens is then passed through a DB lookup procedure which tries to determine the longest match possible. As a result of the process, tokens forming terms are grouped together, and their multiple possible values as proteins are associated to them. As an example, the term **eIF4E** gets 46 different values, such as:

IF4E_ASHGO, IF4E_RAT, IF4E1_SCHPO, IF4EA_BRARE, ..., 4EBP2_HUMAN, 4ET_HUMAN

Although in a few cases the results described in the articles apply to multiple species, in the majority of cases the article focuses on one (or in some cases 2 or 3) organisms.⁶ Being able to determine with precision which is the organism used in the study leads therefore to a huge disambiguation effect.

For our experiments we have adopted a statistical approach based on the occurrences of the mentions of organisms in the various sections of the paper. Just like for proteins, we have stored in our DB a number of well-known synonyms for the organism (e.g. "murine" is an adjective referring typically to "mouse").⁷ The relative frequency of species in the sections of the papers are combined linearly, with weights assigned through a learning procedure over a training corpus, and balanced by the known absolute frequency of species in biological research articles (whereby "human" by far outnumbers all other species). Mentions in the abstract tend to have a predominant role in the balanced statistics.

The algorithm delivers a ranked list of species for each article. Such a list is then used to drastically reduce the number of possible interpretations for each term. The first step of disambiguation (organism-based) will simply go through all values for a term, and select those that match the best ranked organism. If that fails to deliver any result, it will proceed with the next organism, according to the ranking, until an assignment is found, or a given threshold is reached.⁸

Over the Biocreative training data (740 abstracts), the initial annotation step delivers 283556 distinct protein values (P: 0.0072; R: 0.7469).⁹ After the species-based disambiguation step this number is reduced to 45012 (P: 0.0308; R: 0.5763). The remaining ambiguity (intraspecies) needs to be solved by other means.

³<http://www.expasy.org/sprot/>

⁴We used the file "uniprot.light.table.txt", delivered by the organizers at the beginning of September.

⁵To be more precise, we analyzed only sentences contained in the abstracts for the detection of protein interactions, but additional information derived from the full articles was used for one aspect of the problem (organism-based disambiguation).

⁶In the training data, there were 449 articles with interactions involving only 1 organism, 142 articles with 2, 26 articles with 3, 6 articles with 4, 3 articles with 5, 1 article with 6, and 1 article with 9 different organisms (only 628 articles, among those distributed as training data, contained curatable interactions).

⁷Names and synonyms for organisms were automatically downloaded from NEWT (<http://www.ebi.ac.uk/newt/>). HTML pages were parsed using the Java-based open-source HTML parser NekoHTML (<http://people.apache.org/~andyc/neko/doc/html/>).

⁸Currently set at 3, i.e. if an assignment is not found in the 3 best ranked organisms, the term is NOT tagged as a protein.

⁹All P/R/F figures reported in this paper, unless explicitly noted, refer to the **training data**. Due to lack of space and time, a detailed analysis of the results obtained on the test data was not possible. Such an analysis is being conducted and the results will be presented at the BioCreAtIVe workshop.

- The Cap - binding protein eIF4E promotes folding of a functional domain of yeast translation initiation factor eIF4G1 .
 - The association of eucaryotic translation factor eIF4G with the cap - binding protein eIF4E establishes a critical link between the mRNA and the ribosome during translation initiation .
 - This association requires a conserved seven amino acid peptide within eIF4G that binds to eIF4E .
 - Here we report that a 98 - amino acid fragment of *S. cerevisiae* eIF4G1 that contains this eIF4E binding peptide undergoes an unfolded to folded transition upon binding to eIF4E .
 - The folding of the eIF4G1 domain was evidenced by the eIF4E - dependent changes in its protease sensitivity and (1) H - (15) N HSQC NMR spectrum .
 - Analysis of a series of charge - to - alanine mutations throughout the essential 55.4 - kDa core of yeast eIF4G1 also revealed substitutions within this 98 - amino acid region that led to reduced eIF4E binding in vivo and in vitro .
 - These data suggest that the association of yeast eIF4E with eIF4G1 leads to the formation of a structured domain within eIF4G1 that could serve as a specific site for interactions with other components of the translational apparatus .
 - They also suggest that the stability of the native eIF4E - eIF4G complex is determined by amino acid residues outside of the conserved seven - residue consensus sequence .

Figure 1: Example of annotated abstract. The tokens marked in red are those identified by the system as protein names, the tokens marked in blue are those identified as organism names, tokens marked in yellow are indicators for a relation, tokens marked in green might suggest the presence of a curatable relation. The green dot on the left of a sentence indicates that the system considers that sentence as potentially containing a “curatable” relation.

With the collaboration of a domain expert, a small set of rules has been developed, which reflects the typical naming conventions made by the authors. For example, the term **MRGX**, even if we know that it refers to a human protein, is still ambiguous among the following: MRGX1_HUMAN, MRGX2_HUMAN, MRGX3_HUMAN, MRGX4_HUMAN. However, it is a typical convention that, if no further qualifiers are adopted, the term will refer to the first case (MRGX1_HUMAN). Alternatively, where there is a group of proteins characterized by Greek letter suffixes (“-alpha”, “-beta”, etc.), the convention is that the unqualified name usually refers to the “-alpha” variant.¹⁰

By sequentially applying the variant rules suggested by the domain expert, the second disambiguation step typically selects one value for each term. Over our collection of 740 abstract, this reduces the number of possible values to 6351 (P: 0.1311; R: 0.4974). As the figures reveal, one must accept a significant loss of recall at each disambiguation step, in order to reach a minimally satisfactory precision.

3 Identification and selection of Interactions

The training set contains 740 articles obtained from either the INTACT or MINT databases, together with the “gold standard”, i.e. the set of interactions that the curators have identified in each article as novel and relevant (3189 interactions in total). The average number of interactions per article is 4.31, however there are a few articles which contain unusually large number of interactions (the biggest number being 170). According to recommendations by the organizers, we dropped from the training set all articles containing more than 20 interactions. This left 719 articles, of which actually only 628 do contain interactions (for a total of 1900 interactions, average 3.07 interactions per article).

Once reasonable values have been reached in the task of detecting proteins, the next problem to be tackled is that of identifying their possible interactions. A naive approach would simply consist of generating all possible pairs of proteins mentioned in each single abstract. This results in a recall of almost 35%, however at the cost of an abysmal precision.¹¹ Another simple approach consists in enforcing a maximal distance (in number of tokens) between any 2 mentions of the proteins. We have experimented with varying distances from 1 to 50 (without taking into account sentence boundaries), and found the best F-measure value at the distance of 9 (P: 0.0460; R: 0.1765; F: 0.0729).

The conceptually simpler (and more intuitive) approach of restricting the possible combinations to proteins within the same sentence, without requiring any maximal distance, delivers better results (P: 0.0494; R: 0.2077; F: 0.0798).

¹⁰There are a few well-known exceptions, such as “immune interferon” (which is normally used to refer to “interferon-gamma”).

¹¹We decided against submitting such results, although this might have given us better scores for recall, because we think that results with precision inferior to 1% are in any case of little use.

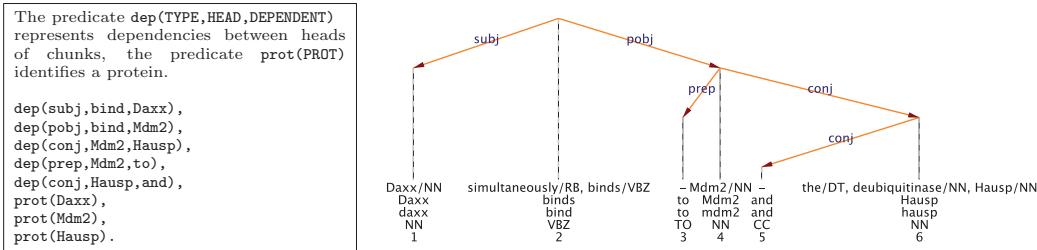


Figure 2: Example of dependency tree (internal representation on the left, graphical visualization on the right)

Still, while recall is relatively good (considering the limitations of the protein detection phase), precision appears too low for a practical application of the approach proposed. Therefore a further filtering phase is required to select among the proposed interactions only those really relevant. In this respect, two kinds of “false positives” need to be distinguished. On the one hand, there are pairs which correspond to interactions mentioned by the authors, but which are not relevant to the curation task, either because they are well-known interactions, or because they play a secondary role wrt the main interactions described. On the other hand, there are genuinely spurious protein pairs, which are not described by the authors as interacting, but are simply a product of the simplistic way in which the pairs are generated. The strategies to filter out the false positives need therefore to address both problems.

In the first case, the approach that we have followed is to try to identify in each abstract the sentences that describe the most relevant results according to the authors, and distinguish them from the sentences that describe background results, an example of which could be the following: *“Previous studies have revealed a genetic interaction between DLG and another PDZ scaffolding protein, SCRIBBLE (SCRIB), during the establishment of cell polarity in developing epithelia.”*

An example of a sentence that reports ‘curatable’ results is the following: *“Here we report the isolation of a new DLG-interacting protein, GUK-holder, that interacts with the GUK domain of DLG and which is dynamically expressed during synaptic bouton budding.”*

In order to distinguish between background and novel information, we adopted a machine learning approach based on a classifier¹² which takes as training data the lemmatized version of sentences whose status has been determined on the basis of the gold standard. A sentence is considered positive if it contains at least one pair of proteins belonging to one of the gold standard interactions for the abstract to which the sentence belongs (see figure 1). After application of the ‘novelty’ filter the results that we obtained on the training data are the following: (P: 0.0945; R: 0.1992; F: 0.1282).

The second problem can be dealt with by taking into account the exact syntactic configuration in which the two proteins appear, i.e. does the context form a meaningful interaction? For example, in the sentence *“Daxx simultaneously binds to Mdm2 and the deubiquitinase Hausp”* three possible interactions can be considered (the direction of the interaction is presently ignored):

1. Daxx – Mdm2 2. Daxx – Hausp 3. Mdm2 – Hausp

However, on syntactic grounds (see figure 2), only the first 2 interactions are licensed, while the third is not justified. We have developed a series of lexico-syntactic filters, which are applied in a cascade to each proposed interaction. The filters make use of lexical, morphological and syntactic information delivered by a pipeline of NLP tools, including a novel dependency parser (for more details see [5]). For example, filters capturing the interactions shown in figure 2 are (using a simplified notation):

```
int(X,Y) ← dep(subj,H,X), dep(pobj,H,Y), prot(X), prot(Y).
int(X,Z) ← dep(subj,H,X), dep(pobj,H,Y), dep(conj,Y,Z), prot(X), prot(Z).
```

Only if at least one of such filters applies to the specific case, the interaction is further considered. The results that we obtain on the training data are (P: 0.5437; R: 0.1839; F: 0.2749). In order to enhance the usefulness and maintainability of the lexico-syntactic filters, a special type of visualization has been created (see figure 3) which shows for each sentence and each interaction potentially therein contained, which filter captures the given interaction.

¹²We used the Rainbow tool (<http://www.cs.cmu.edu/~mccallum/bow/rainbow/>) and tested different methods, obtaining the best results with an SVM approach.

m10067897-s1 UBX_DROME EXD_DROME 32 sf:	<ul style="list-style-type: none"> ● Structure of a DNA - bound Ultrabithorax - Extradenticle homeodomain complex .
m10067897-s6 UBX_DROME EXD_DROME 32 sf:	<ul style="list-style-type: none"> ● The Ultrabithorax and Extradenticle homeodomains bind opposite faces of the DNA , with their DNA - recognition helices almost touching each other .
m10067897-s5 UBX_DROME EXD_DROME gold sf:	<ul style="list-style-type: none"> ● To understand the structural basis of Hox - Extradenticle pairing , we determine here the crystal structure of an Ultrabithorax - Extradenticle - DNA complex at 2.4 Å resolution , using the minimal polypeptides that form a cooperative heterodimer .
m10067897-s5 TLX1_HUMAN EXD_DROME 13 sf:	<ul style="list-style-type: none"> ● To understand the structural basis of Hox - Extradenticle pairing , we determine here the crystal structure of an Ultrabithorax - Extradenticle - DNA complex at 2.4 Å resolution , using the minimal polypeptides that form a cooperative heterodimer .

Figure 3: Support tools for the validation of filters. To the left of each sentence, the target interaction (either from gold standard or derived by the system). Green means the interaction detected by the system matches an interaction in the gold standard. Gold means an interaction in the gold standard not detected by the system. Red means an interaction detected by the system but not contained in the gold standard. In other words, true positives are in green, false positives are in red, and false negatives are in gold.

4 Identification of the Interaction Method

The original idea for this subtask was to compare two methodologies, pattern matching (supplemented by simple statistics) and machine learning. As the resources for this subtask were extremely limited and time was running short, this comparison had to be postponed, so only the results of the pattern matching approach were submitted. Pattern matching was done on a full-text version of the articles, as many abstracts don't mention all methods, nor any hints for them. These are normally mentioned in the "Methods and Materials" section.¹³

The first important decision for this pattern matching approach was that — considering the limited resources and time budget — patterns for most methods could not be written by hand. So we started with the method part of the PSI-MI ontology and took the official names, synonyms and exact synonyms of the methods given there as baseline. These patterns were then supplemented by patterns automatically derived from the baseline patterns by considering several well-known variations such as insertion of spaces and hyphens (everywhere), deletion of spaces or hyphens (between words), interpolation of words (between words), truncation of heads etc. In this phase, just as in the next one, recall improvement was the primary goal.

The selection of methods for which patterns should be written by hand was based on the frequency of the methods in the training data and the recall and precision of the automatically derived patterns. As just 5 methods account for two thirds of all file-method-pairs in the training data, these were carefully looked at by our team's biologist, who tried to find additional hints in some of the papers where the methods were not found by the automatically derived patterns. The method 'coimmunoprecipitation' (MI:0019) and its hyponyms 'anti tag coimmunoprecipitation' (MI:0007) and 'anti bait coimmunoprecipitation' (MI:0006) were most successfully treated that way, because they are extremely frequent in the training data and at the same time seldom recognized by the automatically derived patterns. After identifying files as containing one of the coip methods, the most important problem was the very low precision for most hints with good recall (e.g. "antibod" predicts 'anti bait coimmunoprecipitation' (MI:0006) with recall 0.985 and precision 0.299) and the low recall for most hints with good precision (e. g. "flag-tagged" in combination with "precipitat" predicts 'anti tag coimmunoprecipitation' (MI:0007) with recall 0.434 and precision 0.543).

This could be overcome by a back-off algorithm, starting with the patterns with best precision (assigning their methods and excluding other coip methods), continuing with patterns with a lower precision (assigning their methods non-exclusively) and ending with a default (MI:0019).

Similar approaches for 'pull down' (MI:0096) led to much less improvement because the results for the automatically derived patterns were already rather good. This was even more so for the 5th method, 'two

¹³The full-text version was derived from the HTML-version. As the text files delivered by the organizers were found to be unsuitable, due to the presence of control characters, new files were generated using the command "html2text -nobs". The result is still not ideal for text processing, but definitely better.

hybrid' (0018), so the handcrafted patterns for this method were abandoned.

'Imaging techniques' (MI:0428) was selected for a handcrafted pattern because recall was very bad. It was improved significantly by deriving the new pattern from obsolete method names which have to be mapped to MI:0428 as they don't figure in PSI-MI 2.5 any more. An improvement in precision for 'biochemical' (MI:0401) could be made by coupling the very imprecise pattern with other, more precise hints.

The pattern matching at this stage resulted in about 6.8 candidates per file with good recall (0.734) but bad precision (0.243). Obviously the number of candidates had to be reduced to a degree comparable to the training data. For this, every candidate (method) was given a weight influenced by its frequency in the training data and the precision and recall of the patterns used to detect it.

For the 3 runs to be submitted we decided on the following degrees of reduction: run 1, giving only the best candidate (and so the highest precision), was coupled with the results of the highest-precision-run for subtask 3.2. Run 2, giving the 3 best candidates (for best recall) was coupled with the results of the highest-recall-run for subtask 3.2 and run 3, giving the best F-measure by selecting up to 3 best candidates (additional condition was that candidate 2 and 3 reached a minimum in frequency and precision) was coupled again with the results of the highest-recall-run for subtask 3.2. As the interactants were identified in the abstracts only, whereas the methods were identified in the full text, no direct allocation of methods to specific interactant-pairs could be achieved. So we allocated every method for a file to all its interactant-pairs.

Pattern-matching just on the isolated "methods and materials" chapters of the articles without candidate-reduction had much higher precision than the unreduced pattern-matching of the full text, but after candidate-reduction the results for the full-text pattern-matching were slightly better.

5 Conclusions

This paper presents an approach, directed at the extraction of protein-protein interactions from biomedical literature, based on sequential filtering of candidate interactions (pairs of proteins in sentences). The filters make use of linguistic information derived from a pipeline of NLP tools, in particular including a dependency parser. Further, a pattern-based approach is capable of recognizing the most frequently used experimental methods with a high reliability. The results show that the proposed approach is competitive.

6 Acknowledgments

We thank the anonymous reviewer(s) for their helpful and insightful comments.

References

- [1] Andrew B Clegg and Adrian J Shepherd. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics*, 8:24, 2007.
- [2] Katrin Fundel, Robert Küffner, and Ralf Zimmer. RelEx — Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007.
- [3] Martin Krallinger and Alfonso Valencia. Text-mining and information-retrieval services for molecular biology. *Genome Biology*, 6(7):224, 2005.
- [4] Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Bjorne, Jorma Boberg, Jouni Jarvinen, and Tapio Salakoski. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8:50, 2007.
- [5] Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, Michael Hess, and Martin Romacker. An Environment for Relation Mining over Richly Annotated Corpora: the case of GENIA. *BMC Bioinformatics*, 7(Suppl 3):S3, 2006.
- [6] Gerold Schneider. *Hybrid Long-Distance Functional Dependency Parsing*. Doctoral Thesis, Institute of Computational Linguistics, University of Zurich, 2007.



GeneTeam Site Report for BioCreative II: Customizing a Simple Toolkit for Text Mining in Molecular Biology

Frederic Ehrler^{1,2}

ehrler@sim.hcuge.ch

Julien Gobeill, Imad Tbahriti^{1,3}

{gobeill;tbahriti}@sim.hcuge.ch

Patrick Ruch¹

ruch@sim.hcuge.ch

¹ Medical Informatics Service, University and Hospitals of Geneva

1201 Geneva, Switzerland

² AI Group, University of Geneva

1201 Geneva, Switzerland

³ Swiss-Prot, Swiss Institute of Bioinformatics, Medical Faculty

1201 Geneva, Switzerland

Abstract

In this technical report, we describe our participation in two of the three BioCreative II tasks: gene normalization, article selection for protein-protein interaction and protein-protein interactions. We report on the customization of a simple modular toolkit, which can be applied several text mining applications in molecular biology. The toolkit comprises an automatic generic text categorizer, a retrieval engine and an argumentative classifier, trained to differentiate between PURPOSE, METHODS, RESULTS and CONCLUSION in MEDLINE abstracts. The automatic text categorizer requires a very limited tuning set, and the system keeps most of its effectiveness when tuning data are sparse. We use the categorizer for several subtask: Gene Normalization of ENTREZ-Gene entries, selection/ranking of relevant articles, recognition of Swiss-Prot protein identifier. This last task assumes that we are able to: recognize species, select appropriate sentences, and finally be able to automatically assign interaction detection methods. The overall results, although still partial at the time of writing this report, show that our toolkit can achieve competitive performances with minimal task customization efforts.

Keywords: text mining, text categorization, protein-protein interaction, database curation, machine learning, information retrieval

1 Introduction

In this report, we describe the participation of the GeneTeam (Geneva Team, participant no 14) for the BioCreative II initiative. For this second participation, we participate in two tasks: Gene Normalization and Protein-Protein interactions. The Protein-Protein interaction task is separated into four subtasks: article selection (IAS), identification of pairs of proteins (IPS), sentence selection (ISS) and methods extraction (IMS). We first describe the methods used and the results reported for the Gene Normalization (GN), then we detail the Protein-Protein interaction (PPI) task and related results, whose numerous and challenging subtasks have absorbed most of our participation efforts this year. The background section presents some of the generic tools, which were used for the different tasks and subtasks, then we report on developments or customization needed to achieve the tasks.

2 Background

In this section, we briefly introduce two tools, which are widely used in our experiments: a generic data-poor biomedical text categorizer and an argumentative classifier, which classifies each sentence into a four-class rhetorical model: PURPOSE, METHODS, RESULTS, CONCLUSION. Like for the

first edition of BioCreative, we limit all our investigations to abstracts and do not attempt to take advantage of full-text articles. Indeed, the free availability of full-text articles in HTML format is marginal compared to the 16 millions abstracts freely available in MEDLINE.

2.1 Generic categorizer

An important module in our architecture is given by a general automatic text categorization framework. The framework is adapted to handle large multi class classification problems [1]. In [2], the system is also applied to automatically annotate Swiss-Prot proteins with Gene Ontology categories in the context of the BioCreative I challenge. In [1] the system is applied for keyword assignment of Medical Subject Headings to MEDLINE abstracts.

Unlike usual automatic text categorization systems, which rely on data-intensive models extracted from large sets of training data, our categorizer is largely data-independent and a small sample is sufficient to tune the system.

Following previous observations made in BioCreative I, we decided this year again not to use the full-text contents of articles but instead to concentrate our mining efforts on abstracts. Together with abstracts, the title and other fields of MEDLINE records (MH and RN fields) are used to generate our runs. Each article is augmented with automatically assigned Medical Subject Headings when these keywords are not already provided in MEDLINE.

Our categorizer is based on two ranking modules: a pattern matcher and a vector space retrieval engine. The vector space engine (the easyIR toolkit; cf. [3]) must be parameterized to obtain the best weighting schema. The combination of the pattern matcher and the retrieval engine must also be parameterized. For our experiments, a slightly modified dtu.dtn [4] formula (i.e. term frequency, document frequency and pivoted normalization) was selected based on previous experiments. The categorizer outputs a score, which computes a linear combination between the retrieval status value of the retrieval engine, the maximal length of the matching category, and the number of matching features (Boolean scoring). It uses both stems and linguistically-motivated indexing units, in particular noun phrases. A simple S-Stemmer [5], which handles plural English forms was used in all our experiments. A list of stop words is also needed, as well as a list of stop categories. Stop words are removed before categorization, while stop categories are removed after categorization.

2.2 Argumentative classifier

The argumentative classifier [6] [7] [8] merges a Bayesian learner and a hidden Markov model to categorize each sentence in four argumentative classes: PURPOSE, METHODS, RESULTS, CONCLUSION. Following observations made in [9] for automatic assignment of Medical Subject Headings and in [10] for extraction of GeneRiFs (Gene Reference into Functions), we overweight features appearing in sentences classified as PURPOSE, as compared to features appearing elsewhere in the abstract. This strategy is applied to all tasks requiring text categorization: Gene Normalization, assignment of Medical Subject Headings, assignment of species names, which is used for identification of Swiss-Prot Accession Numbers.

3 Gene Normalization

Gene lists were provided by the organizers. Because we obtained the tuning data a few hours before the deadline, we borrowed the settings of the categorization system from the MeSH categorizer [1]. Fortunately, we were still able to use the tuning data to establish a specific list of stop words and stop categories. These lists were augmented using differential frequency sets established on biomedical (TREC Genomics collection 2004 [11]) and journal (Wall Street Journal) corpora.

	GN1	GN2	GN3
Precision	0.762	0.471	0.479
Recall	0.485	0.483	0.655

Table 1: Results for each official run.

For this task, the resulting output of the categorizer is a ranked list of categories with a normalized score for every category. The number of categories per article ranges from 0 (no category was detected) to 15.

As demanded by the task definition, for every category, we attempt to recover the gene name as found in the text. This passage recovery is based on the computation of a string-to-string edit distance between the predicted category and the abstract or the title. Two different thresholds are tested for the computation of the edit-distance (Table 1): exact similarity (GN1), one edit-distance (GN2), two-edit distance (GN3). In order to follow the BioCreative II Gene Normalization protocol, categories which are relatively distant from textual strings, i.e. beyond two-edit distances, are simply removed. Although Medical Subject Headings fields do contain relevant gene names, they are ignored when they do not appear in the abstract. BioCreative guidelines assume that gene normalization is to be performed on full-text articles, while in our experiments we restrict our passage search to the abstracts. Therefore it is expected that effectiveness of the categorizer could be significantly improved by recovering strings in the full-text as well.

The main parameter of the experiment is given by the string-to-string edit distance module, which can use different threshold to accept more matching strings. As expected, relaxing matching constraints results in trading precision for recall as shown between runs GN1 and GN3. If textual representatives were not demanded -thus transforming the task in a categorization task without exploring full-text- significant improvements could have been expected. From a strict user perspective it is unclear how the design of task can correspond to some real needs, in particular if we consider that genes names are fairly ambiguous with respect to species. Thus, while the protein-protein interaction (see below) tasks was legitimately addressing species disambiguation issues, this parameter was somehow artificially neutralized for the Gene Normalization task.

4 Protein Protein Interaction and subtasks

The Protein-Protein interaction task is separated into several subtasks: article selection (IAS), identification of pairs of proteins (IPS), sentence selection (ISS) and methods extraction (IMS).

4.1 Article Selection

This subtask of the protein-protein interaction task is formally defined as a binary classification problem, but in addition to this classical problem, the evaluation protocol demands to transform the classification problem in a ranking problem. Because large training data sets are available, we decided to rely on well-established learning methods. To achieve the task: three steps are required: the choice of the classification algorithm, the selection of relevant features, and the ranking strategy.

4.1.1 Feature selection and weighting

The features selected to represent the documents should contain enough information to discriminate correctly the documents that contain protein interactions from the others. One possible approach could have been to select all the words that compose the documents as features. However we observed that given the high number of protein variants, such features will not possess the generalization property required for the classification process. Therefore instead of using words, we tried to find others features that still reflect the content of the documents but that allow generalizing more efficiently the important concepts contained in the text. MeSH (Medical Subject Headings) categories were primarily used as features as they seem to synthesize our requirements. First, assigned MeSH categories are both reliable and somehow consistent as they are provided by expert librarians. Second, they summarize the key concepts of the documents. Third, they are less numerous than the words and thereby create a classification problem of lower dimensionality. In order to find the MeSH terms related to the documents, we used a tool that is able, given a PMID identification number, to query PubMed in order to fetch the related MeSH terms. For MEDLINE records without MeSH categories, these categories were generated by automatic text categorization [1]. Even if using MeSH terms as features create a space of lower dimensionality, the number term associated with the collection of documents is still fairly high to be handled for most advanced learners, therefore we perform a selection of features. Reducing the feature space is done using information gain. The gain of entropy of each feature is computed in order to keep only the features that bring sufficient information to achieve a decision. Applying this selection step allows to reduce the initial set of features by 90%.

Although MeSH features seem appropriate to represent the content of the documents, we decided to add dedicated features that are especially discriminative for our protein-protein interaction task. In particular, occurrences of interaction verbs are used to discriminate the two classes. A thesaurus, containing all interaction verbs, was manually generated. Then, interaction verbs are extracted from the abstracts using simple pattern matching method. In the same vein, we decided to use the protein names as features. However, as proteins names are too specific, we have not used them directly as features. Using each of them as separate features will have been meaningless as there is very little chance to find two similar protein names in two different articles. So instead of using directly the protein names, we simply used the number of protein names found in the text as feature. In order to extract the features, we used again pattern matching technique. Proteins names, as found in the GPSDB, are searched in the documents. GPSDB (Gene and Protein Synonym DataBase) is a collection of gene and protein names organized by species. Two different methods have been explored to retrieve from the abstracts the protein names contained in GPSDB. The first and most conservative approach consists in an exact match applied to the abstracts, while in the second we consider as protein every word that haven't been found in a common English thesaurus. The words contained in the thesaurus used on this stage are extracted from the Wall Street journal. The choice of the algorithm has been done based on the performance obtained on the training data. The test of several algorithms known to perform correctly with textual data leads us to select SVM (Support Vector Machines). Linear kernels are particularly appropriate as they directly provide a weight for every feature. These weights can then be combined on every document to obtain a linear score, which will be used for ranking. For the ranking step, decision boundaries were recalculated by computing density estimation of errors in the neighborhood of the boundaries.

4.1.2 Results

The three submitted runs were generated using different features, cf. Table 2. For the first run, we used MeSH terms and interactions verbs (IAS1). In the second run we used MeSH terms, interactions verbs and the number of proteins names extracted using the GPSDB direct matching technique (IAS2).

In the last approach we used again MeSH terms, interactions verbs and the number of extracted proteins but this time the number of protein is inferred from the number of words which are absent

	PRECISION	RECALL	F-SCORE	AUC
IAS3	0.75	0.512	0.61	0.766
IAS2	0.74	0.504	0.6	0.764
IAS1	0.74	0.49	0.59	0.761
Median	0.677	0.85	0.72	0.751

Table 2: Results for each official run, compared to the median.

from a common English words list (IAS3) before being compared with those in GPSDB. Although differences are probably not significant, considering any out-of-vocabulary word as a potential protein name seems slightly more effective than using directly a knowledge-intensive resource such as GPSDB. In general, using protein-related features has anyway a limited impact when compared to using simply Medical Subject Headings. Our results are generally slightly above the median, with a relative strong precision and lower recall.

4.2 Detection of proteins pairs, sentences and protein interaction methods

We gather in this section all remaining subtasks of the protein-protein interaction task.

The first step consists in detecting in the text the interaction verbs and the protein names. In order to identify the interaction verbs, we first create a thesaurus that should contain all the verbs that indicate the presence of an interaction between several proteins. To build such resource we have manually parsed the training data to retrieve the verbs that appear conjointly with the proteins participating to an interaction. Once the words locating the interactions extracted, we need to extract the protein names. To solve this problem, we decided to apply different matching techniques to retrieve in the abstracts the words contained in a predefined list of proteins. Protein names come from GPSDB (Gene and Protein Synonym Database) a collection of gene and protein names organized by species. At this level several matching approaches have been tested depending of the desired flexibility. The first and most conservative approach consists in searching directly in the abstracts the occurrences of the protein names existing in GPSDB.

For the second approach, we first filter out from GPSDB all the words which are not specific to proteomics and then perform the matching. As we keep only the key terms of every protein, we increase the number of matches. On the third approach we try again to offer flexibility in the matching process by allowing variations of protein names (case, hyphen and parenthesis). As the number of existing protein names is large, it will be very time consuming to search the abstracts for every variation of protein names, therefore, to reduce the number of required comparisons, we first remove from the abstracts all the words existing in a common English thesaurus. The thesaurus is built from frequency lists computed on the Wall Street journal. Once the abstracts is cleaned we apply a pattern based matching technique that allows variations in the spelling of the protein names.

4.2.1 Interaction detection

Once the protein names and the interaction verbs are extracted from the abstracts, we have to decide which proteins participate to a given interaction. During this step we must avoid to create irrelevant interactions by linking unrelated proteins. To extract the interactions, we first split the abstracts into sentences using a rule-based sentence splitter. Once done, we look for all possible interactions in every sentence, which contains at least two protein names and one interaction verb. The choice of the

interacting proteins is basic if the analyzed sentence contains only two proteins. However, in most of the case we find more than two proteins in the sentence, therefore we have to select which ones are related. To rank these hypothetical interactions we compute a distance between each proteins pair and the interaction verb. In the current state of the system, we limit the scope of the interaction search to intra-sentences pairs, although anaphoric phenomena [12] could demand more elaborated search strategies. This should be particularly true in full-text contents.

4.2.2 Species categorization

Knowing only the proteins names is not sufficient as several proteins can share a similar name in several species. Therefore we need to link each protein to a species in order to solve inter-species ambiguities. For disambiguating species, we rely again on our automatic text categorization framework. Species categories are provided by NEWT, but we also had to define a short mapping table between textual entities (e.g. *mouse*), Medical Subject Headings (e.g. *mice*), NEWT (*mus musculus*) and Swiss-Prot, with about 20 entries. The number of species returned per article ranges from 0 (no category was detected) to 5.

4.2.3 UniProt ID recovery

We have to provide the UniProt ID of every protein participating to an interaction. For every hypothetical interaction pairs, we obtain the UniProt ID by crossing the species and the protein name. When the conjunction of the list of species and of proteins is an empty set, we backtrack and assume that the appropriate species is not available, so we search which species could be related with the maximum number of interacting proteins. Once every protein is assigned to a set of species, we can attempt to map each pair of {protein; species} to a group identification number of GPSDB.

Run (maximizing the F-score)	Mean
Precision total interactor protein-article associations:	0.277904328018 0.0938
Recall total interactor protein-article associations:	0.186830015314 0.1064
F-score total interactor protein-article associations:	0.223443223443 0.0781

Table 3: Results of our optimal run regarding F-score for the interactor protein normalization subtask, compared to the mean. Several metrics were proposed by the organizers but again they tend to correlate.

Results of the protein pair normalization are given in Table 3. Although our results are significantly better than the mean, it is observed that the absolute performances are still modest.

4.2.4 Interactions and sentence selection

The documents used for the protein interaction pair generation subtask are those classified as relevant during the classification task (IAS). Knowing the positive documents of the previous task allow us to build a classification model able to classify all the IAS documents. This model makes possible a ranking of all the documents with regard to their probability to contain protein interactions. The interactions of proteins are then ranked in agreement with the source document. When we are in a situation where there exists more than one interaction for a document, we have to rank the interactions

	IPS1	IPS2	IPS3
MRR	0.8718	0.8167	0.8718
Mean	0.1062	0.1858	0.1035

Table 4: Mean reciprocal rank (MRR) of correct passage for each official run. Several metrics were proposed with high correlations so we report only the average precision, recall and f-score for extracting the normalized protein interaction pairs corresponding to SwissProt entries for each article.

[TITLE interacts with Lyn and is critical for erythropoietin-induced differentiation of erythroid cells.]
CONCLUSION (00160116) These data show that disrupting HS1 has profoundly influenced the ability of erythroid cells to terminally differentiate.
CONCLUSION (00160055) The truncated HS1 also suppressed the development of erythroid colonies from fetal liver cells.
CONCLUSION (0015972) The inability of cells containing the truncated HS1 to differentiate may be a consequence of markedly reduced levels of Lyn and GATA-1.
CONCLUSION (0015830) In addition, erythropoietin stimulation of these cells resulted in rapid, endosome-mediated degradation of endogenous HS1.
METHODS (00162303) A truncated HS1, bearing the Lyn-binding domain, was introduced into J2E erythroleukemic cells to determine the impact upon responsiveness to erythropoietin.
PURPOSE (00176456) Previously we have shown that the tyrosine kinase Lyn associates with the erythropoietin receptor and is essential for hemoglobin synthesis in three erythroleukemic cell lines.
PURPOSE (00167817) Here we show that the hemopoietic-specific protein HS1 interacted directly with the SH3 domain of Lyn, via its proline-rich region.
PURPOSE (00154385) To understand Lyn signaling events in erythroid cells, the yeast two-hybrid system was used to analyze interactions with other proteins.
RESULTS (00155338) Truncated HS1 had a striking effect on the phenotype of the J2E line-the cells were smaller, more basophilic than the parental proerythoblastoid cells and had fewer surface erythropoietin receptors.
RESULTS (00155011) Moreover, basal and erythropoietin-induced proliferation and differentiation were markedly suppressed.

Table 5: Example of argumentative classification output for the abstract PMID no 10713104. For each row, the assigned argumentative class is followed by the score for the class, followed by the extracted text segment .

within the document. For this purpose we look at the number of occurrences of any given interaction in the document and we rank them based on their respective frequencies.

As for the extraction of the interaction pairs we use a technique that retrieves the interactions at the level of sentences. Indeed, we already know which sentences support the interactions, therefore, retrieving the sentences related to a given interaction does not require any additional work. However, there are cases where several sentences are relevant for an interaction. In such situation, we must rank the sentences: the strategy consists in counting the number of time they support the given interaction. The rank is computed based on the frequency of citations.

4.2.5 Protein Interaction Method, Results and Discussion

For assigning method interaction methods, we again use the same tool as the one used for species disambiguation, but instead of using species categories, as provided by NEWT, we used an OBO resource, provided by the organizers. Due to a lack of time, probably suboptimal tuning parameters were simply borrowed from settings established for Medical Subject Headings.

It is worth observing that in our best run (IM1), features appearing in the METHODS and PURPOSE section of the abstracts are simply overweighed (x2), as compared to other argumentative contents; see [9] for a description of such a rhetorical boosting approach for automatic assignment of keywords in MEDLINE. The boosting factor was set a priori and additional experiments will be needed to establish the effectiveness of argumentative representation levels for such a task. An example of the output of the argumentative classifier is given in Table 5. In this example, we observe that the passage

CSV	CSV-B	Category	ID
0020095	0020238	competition binding	0405
0020095	0020238	saturation binding	0440
0020095	0020238	filter binding	0049
0026454	0029738	three hybrid system	0438
0043567	0047459	two hybrid	0018

Table 6: Assignment of protein interaction methods for the PMID no 10713104 (cf. Table 5). Column 2 gives the CSV (Categorization Status Value) before argumentative boosting. Column 3 gives the same value after argumentative boosting, i.e. after applying a multiplicative factor on the term frequency of the argumentatively-selected sentences. Column 3 gives the category, while column 4 indicates the PSI-MI identifier.

	AvPrec	AvReca	AvFsc
IM1	0.3628	0.2172	0.2513
IM2	0.3186	0.1980	0.2249
IM3	0.3348	0.1938	0.2265
Best	0.5068	0.5222	0.4836

Table 7: Results for each official run, compared to the best run (participant T40). Alternative metrics were proposed, in particular metrics accepting as positive categories which are hierarchically close in the PSI-MI hierarchy, they do correlate with exact match metrics.

describing the interaction method (*two hybrid; PSI-MI 0018*) is classified as a PURPOSE rather than as METHODS, which justifies the overweighing of these two argumentative categories. Table 6 shows the impact of argumentative boosting on the automatic assignment of protein interaction methods using the same example.

5 Conclusion

It is premature to draw any final conclusion from these partial results but some trends can be observed. Thus, we can observe that both Gene Normalization and assignment of Protein Interaction Methods tasks should significantly be improved by using full-text contents. While tasks, such as article selection and protein-protein interactions should be neutral with regards to using full-text or abstracts. Additionally, we can observe that knowledge-intensive resources, such as gene and protein thesauri need specific developments to be exploited with effectiveness.

Regarding the use of our ready tools, we can observe that our simple framework (a categorizer, an argumentative classifier and a passage retrieval engine), which have been developed without targeting any particular competition can be appropriately customized for a wide variety of text mining applications: from BioCreative I [13] to TREC Genomics (e.g. [3]) to BioCreative II. This was achieved within a fairly limited time frame if we consider that no more than four full-time equivalent man weeks were allocated for all the tasks, while results seem fairly competitive. Furthermore, the absence of tuning data for several subtasks, did not seem to affect significantly the effectiveness of our architectural choices. In general the scalability, flexibility and customization power of the current tools seem sufficient to address a wide range of ad hoc and heterogeneous tasks.

Finally, we also can observe that our somehow arbitrary decision to limit our investigations to the abstracts of scientific articles did not seem to affect the performance of our tools, at least regarding complex tasks such as the detection of relevant protein interactions or article selection. This result is consistent with previous conclusion drawn from BioCreative I.

Acknowledgements

The list of interaction verbs was provided by Dietrich Rebholz-Schuhmann from the EBI. The GPSDB resource was kindly shared by Anne-Lise Veuthey from the Swiss-Prot group of the SIB. We also would like to thank the organization team, and particularly Martin Krallinger for its responsiveness. The study was supported by the SNF (3252B0-105755).

References

- [1] Ruch P: **Automatic Assignment of Biomedical Categories: Toward a Generic Approach.** *Bioinformatics* 2006, 6.
 - [2] F Ehrler AG A Jimeno Yepes, Ruch P: **Data-poor Categorization and Passage Retrieval for Gene Ontology Annotation in Swiss-Prot.** *BMC Bioinformatics* 2005, 6 (suppl. 1).
 - [3] Aronson A, Demner-Fushman D, Humphrey S, Lin J, Liu H, Ruch P, Ruiz M, Smith L, Tanabe L, Wilbur J: **Fusion of Knowledge-intensive and Statistical Approaches for Retrieving and Annotating Textual Genomics Documents.** In *TREC 2005*. 2006, .
 - [4] Singhal A, Buckley C, Mitra M: **Pivoted document length normalization.** *ACM-SIGIR* 1996, 21–29.
 - [5] Harman D: **How effective is suffixing ?** *JASIS* 1991, 42 (1):7–15.
 - [6] Ruch P, Baud R, Chichester C, Geissbühler A, Lisacek F, , Rebholz-Schuhmann D, Tbahriti I, Veuthey A: **Extracting Key Sentences with Latent Argumentative Structuring.** In *Int J Med Info.* 2007, 835–40.
 - [7] Tbahriti I, Chichester C, Lisacek F, Ruch P: **Using Argumentation to Retrieve Articles with Similar Citations: an Inquiry into Improving Related Articles Search in the MEDLINE Digital Library.** *International Journal of Medical Informatics* 2005, to appear.
 - [8] Ruch P, Gobéill J, Tbahriti I, Aronson A: **Argumentative feedback: A linguistically-motivated term expansion for information retrieval.** *ACL* 2006, 286–293.
 - [9] Ruch P, Gobéill J, Tbahriti I, Lisacek F, Veuthey A, Aronson A: **Using discourse analysis to improve text categorization in medline.** *MedInfo'07 Proceedings* 2007 (to appear), .
 - [10] Ruch P, Perret L, Savoy J: **Features Combination for Extracting Gene Functions from MEDLINE.** In *European Colloquium on Information Retrieval (ECIR)*. 2005, 112–126.
 - [11] Hersh WR: **Report on the trec 2004 genomics track.** *SIGIR Forum* 2005, 39:21–24.
 - [12] Strube M, Hahn U: **Functional centering.** In *Proceedings of the 34th annual meeting on Association for Computational Linguistics.* Association for Computational Linguistics, 1996, 270–277.
 - [13] Hirschman L, Yeh A, Blaschke C, Valencia A: **Overview of BioCreAtIVe: critical assessment of information extraction for biology.** *BMC Bioinformatics* 2005, 6 (suppl. 1).



AKANE System: Protein-Protein Interaction Pairs in the BioCreAtIVe2 Challenge, PPI-IPS subtask

Rune Sætre¹

satre@is.s.u-tokyo.ac.jp

Kazuhiro Yoshida¹

kyoshida@is.s.u-tokyo.ac.jp

Akane Yakushiji²

yakushiji.akane@jp.fujitsu.com

Yusuke Miyao¹

yusuke@is.s.u-tokyo.ac.jp

Yuichiro Matsubayashi¹

y-matsu@is.s.u-tokyo.ac.jp

Tomoko Ohta¹

okap@is.s.u-tokyo.ac.jp

¹ Department of Computer Science, University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033, Japan

² FUJITSU LABORATORIES LTD. 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8588, Japan

Abstract

This report summarizes the participation of the Tsujii-lab group in the 2006 BioCreative2 text mining challenge¹. It describes the systems used, the results attained, and the lessons learned. The basic idea was to see how well the AKANE system could perform on a full-text Protein-Protein Interaction (PPI) Information Extraction (IE) task. AKANE system is a recently developed, sentence-level PPI system that achieved a 57.3 F-score on the AImed corpus. In order to use the AKANE system for the BioCreative task, the given training data had to be preprocessed. The BioCreative training data contained just a list of interacting protein pair identifiers for each given full-text article, while the expected input for the AKANE system is annotated sentences like in the AImed corpus. In order to transform the full-text articles into AImed sentence-level annotations, the text was first stripped of all HTML coding to get a plain text representation. Then, each mention of protein names were tagged by a Named Entity Recognizer (NER), and all interacting and co-occurring pairs in single sentences were used for training. A pipeline architecture was made to deal with each of these challenges. Some postprocessing was also necessary, in order to transform the results from the AKANE system into the expected format for the BioCreative2 challenge. The postprocessing included filtering and ranking the results, and balancing precision and recall to maximize the F-score.

Keywords: bionlp, protein-protein interaction, natural language processing

1 Introduction and Methods

Our system implements a pipeline architecture, where the modules deal with Sentence Detection (SD), Named Entity Recognition (NER), Parsing, and Protein-Protein-Interaction (PPI) extraction. All the modules use machine learning to maximize the performance on small manually annotated biological training corpora. A separate system was made for transforming the article level BioCreative training data into a sentence level AImed PPI-style format (See section 1.3.1). Each module is briefly described below.

1.1 Sentence Detection

The sentence splitter for biomedical text was trained by a maximum entropy (MaxEnt) method [1], and it employs the GENIA corpus for training [4]. First, the sentence splitter detects candidate

¹http://biocreative.sourceforge.net/biocreative_2.html

positions for splitting using selected delimiters: periods, commas, single/double quotation marks, right parentheses, etc. Then, it classifies whether the positions really split sentences or not. Features used by the classifier are: Delimiters, Previous/Next words, and info about special characters, numbers and capitalization. Some transformations of the words, like removing commas, parentheses, etc. and making lower-case versions were also used. The classifier achieved an F-score of 99.7 on 200 unseen GENIA abstracts. However, there seem to be slightly more errors on the BioCreative data set, mainly because of full text HTML encoding and figure explanation texts.

1.2 Named Entity Recognition

The named entity identifier takes sentence-split, POS-tagged sentences as input. It first applies a statistical named entity recognizer to the input. The statistical recognizer was trained on the data provided by the JNLPBA [5] shared task for named entity recognition. The named entity recognizer outputs marginal distributions of the probability that a substring of the sentence is a protein name. The substrings that have probabilities above some threshold are taken as protein candidates. Then such candidates are mapped to dictionary items whose string edit distance from the candidates are less than some threshold. IDs were taken from Uniprot augmented with the GENA dictionary[6].

When a name is ambiguous, a MaxEnt classifier is used to rank the candidate IDs. The classifier is trained on 296 articles from the training data, using the following features: Similarity between the target article and the MEDLINE articles which is referred to by the Uniprot entry; Similarity between the target article and the MEDLINE articles which include the organism name which is specified by the Uniprot entry; Source dictionary (Uniprot/GENA); Edit distance of the dictionary item and the target string; and Type of the dictionary item (e.g. protein name, gene name, etc.). Similarity was estimated by the cosine measure of the articles represented by tf-idf vectors. Probabilities assigned to each ID by the MaxEnt classifier are output and used by the filtering module (see end of 1.3.1).

1.3 AKANE System

For doing the actual protein pair extraction, the AKANE system [7] was used. It requires AImed Corpus style [2] input for training, so a preprocessor was made to automatically create this kind of co-occurrence sentence collection for the interacting proteins.

The AKANE system parses the input text using the Enju HPSG parser for bio-English. Although the parser has been trained with newswire articles, i.e. Penn Treebank, it can compute accurate analyses of biomedical texts owing to our method for domain adaptation, using the GENIA Treebank [4] to adjust the parsing model. The evaluated bio-performance is 86.9 F-score [3]. The AKANE system combines the output from the parser with the protein pair info from NER, to create the smallest connected parse tree (raw pattern) that covers both proteins. Extra new patterns are also generated by recombining the parts of the raw patterns. Then, counting is done on the training corpus, to evaluate how accurate the patterns are in predicting (only) true interactions. The output from AKANE system lists all possible interactions, so for one mention of an interaction in the text, several interactions are suggested. This is because each protein name is usually ambiguous among several candidate protein IDs, so a postprocessor was made to pick (only) the most likely interaction pair, based on NER probabilities. This is better explained at the end of the following subsection, about pre- and post processing.

1.3.1 Training Data Generation and Pair Filtering based on NER scores

All sentences containing two or more proteins from the PPIs given in the training data files from BioCreative were extracted, and transformed into an AImed style XML marked-up corpus that could be used to train the AKANE system. We assumed that all sentences with a co-occurrence of two interacting (according to the training data from BioCreative) proteins really were describing that

interaction. The accuracy of this assumption, and the effect it had on the prediction phase, was not properly measured (due to lack of time), but some manual inspection of the created corpus indicated reasonable accuracy. Another problem was that only 250 of the total 740 training articles could be used for training. The reason for this is that the AKANE system did not scale well to the large amount of text, compared to the much smaller AImed corpus. So we decided to use only the co-occurrence sentences, and only from the articles where all interacting protein names/IDs could be recognized by NER. Some articles with too many co-occurrence sentences were also dropped, because of the scalability bug in our system.

In order to deal with ambiguity, only the single most likely protein ID were picked from any fragment of ambiguous text, and only the 20 most likely PPI pairs (based on multiplying the NER probabilities) for each article were reported. In run number 2 and 3, a filter was made to remove all pairs that did not have identical species tags in the last part of their protein identifiers. For example, a suggested interaction between P19235 (epor_human) and Q62225 (cish_mouse) is filtered away.

2 Results and Discussion

The three runs were made as follows: Run1 is a version of the system not using the inter-species interaction filter. It achieved an overall F-score of 10.5 (P:8.2% and R:14.6%). Run2 was the best run in terms of F-score based on the training set. On the test data it achieved an overall F-score of 13.7 (P:10.6% and R:19.1%). Run3 was the original AKANE system, trained with the AImed corpus, and optimized for best F-score on the training set. We did not have time to use the machine learning component of AKANE system (F-score 57.3), so instead we used manually tuned parameters and a threshold value reported to achieve 42.0 F-score on AImed (P:70% and R:30%). Still, in the evaluation, Run3 was actually the best one, with an overall F-score of 15.8 (P:15.7% and R:15.9%). This means that training on full text co-occurrence training sentences did not perform any better than training on AImed abstracts alone. The reason for this is that the automatic generation of the training corpus included some noise, in terms of “interacting” co-occurrence like the sentence: *A and B were bought from Santa Cruz inc.*

References

- [1] Berger A.L., Pietra S.D., and Pietra V.J.D., A maximum entropy approach to natural language processing, *Computational Linguistics*, 22(1):39–71, 1996.
 - [2] Bunescu R.C. and Mooney R.J., Subsequence kernels for relation extraction, in *NIPS*, 2005.
 - [3] Hara T., Miyao Y., and Tsujii J., Adapting a probabilistic disambiguation model of an HPSG parser to a new domain, in *IJCNL 2005*, vol. 3651 of *LNAI*, 199–210, Springer-Verlag, Jeju Island, Korea, October 2005.
 - [4] Kim J.D., Ohta T., Tateishi Y., and Tsujii J., GENIA corpus - a semantically annotated corpus for bio-textmining, *Bioinformatics*, 19(suppl. 1):i180–i182, 2003.
 - [5] Kim J.D., Ohta T., Tsuruoka Y., Tateishi Y., and Collier N., Introduction to the bio-entity recognition task at JNLPBA, in *Proceedings of the JNLPBA-04*, 70–75, Geneva, Switzerland, 2004.
 - [6] Koike A. and Takagi T., Gene/protein/family name recognition in biomedical literature, in *Proc. Biolink 2004*, 9–16, 2004.
 - [7] Yakushiji A., *Relation Information Extraction Using Deep Syntactic Analysis*, Ph.D. thesis, University of Tokyo, 2006.



Consensus pattern alignment to find protein-protein interactions in text

Jörg Hakenberg¹

hakenbergj@biotec.tu-dresden.de

Michael Schroeder¹

ms@biotec.tu-dresden.de

Ulf Leser²

leser@informatik.hu-berlin.de

¹ Biotechnological Centre, Technische Universität Dresden, 01307 Dresden, Germany

² Computer Science Department, Humboldt-Universität zu Berlin, 10099 Berlin, Germany

“Don’t I know you from somewhere?” – comparing new to known texts plays a key role in the system we propose for searching protein–protein interactions (PPIs). Our system builds on an inexact pattern matching strategy, where patterns (linguistic frames) reflect the compositional structure of known occurrences of PPIs in text. To describe this structure, part-of-speech tags (verbs etc.) and entity classes (proteins), words, and word stems are used. Consider the sentences “Sky1p phosphorylates Npl3p” and “Akt phosphorylates beta-catenin”. Both have a structure in common that connects two proteins with a single verb. From comparable systems proposed before [1, 2], it became clear that collecting a suitable set of patterns is of major importance, and this step forms the main component of our system. From the IntAct database [5], we extract all pairs of proteins known to interact. We scan PubMed for textual evidences for each such interaction, and retain all single sentences that describe them. Using pairwise sentence alignment as a similarity scoring function, we perform a clustering on the resulting set of sentences. Within each cluster, multiple sentence alignment (MSA) identifies commonalities and variable positions across all sentences, expressed in a consensus pattern. Figure 1 shows an example MSA with four sentences that define one consensus pattern. We can now align such consensus patterns against arbitrary text to extract new PPIs.

Our system yields a maximum recall of 69% –which was the best reported among all participating systems–, a maximum precision of 45% and maximum F1-measure of 41% on the BioCreative test set. Our method works completely independent from the training corpus, which we did not use at any stage. Thus, we intrinsically exclude any risk of overfitting, and believe that our approach should work equally well for related extraction problems, such as finding protein–disease associations.

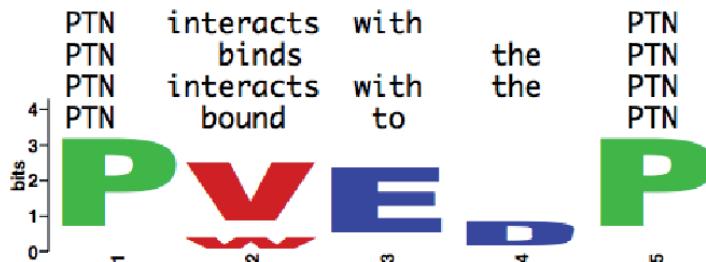


Figure 1: Sequence logo –the consensus pattern– for four short sentences. Height of a character corresponds to the information content (entropy) at this position; the larger a character, the more conserved it is across multiple sentences. PTN/P are wildcards for any protein name; V, verb, present tense; W, verb, simple past; E, preposition; D, determiner. Logo created with WebLogo (weblogo.berkeley.edu).

Methods

The system we propose falls into two components: searching sentences that contain two identified proteins and searching for PPIs described in these sentences. The initial recognition of protein names is based on a dictionary derived from UniProt (protein names, gene names and respective synonyms). The identification is a variation of the system we presented for the GN task (see elsewhere in this proceedings). The extraction of PPIs builds on ideas presented with the Ali Baba tool [3].

Named entity recognition

For the initial recognition of protein names, we built a dictionary using synonyms provided by UniProt/TrEmbl (description and gene name fields) for the approximately 200,000 proteins listed in the IPS data set. Each synonym mapped to all UniProt identifiers that share this synonym. Multiple IDs appeared mainly for abbreviations, which often have different expansions, and proteins shared across multiple organisms. We added term variations (plural/singular forms, changes in capitalization, structural variations) to this dictionary. It was also very important to expand the list of candidate IDs by cross-checking for proteins sharing similar synonyms in UniProt. For example, UniProt contains the name “Hoxb6” only for a protein in the mouse, and uses the name “HOXB6” for human and others. From the training data it became clear that authors often would use “Hoxb6” to refer to the human ortholog, however. Thus, we iteratively expanded the list of IDs for each name variant based on case-insensitive comparisons. We finally compiled a finite state automaton from all entries for fast spotting of candidate names in text.

Named entity normalization

Named entity normalization (NEN) was a very important step in the IPS task, and a proper protein name disambiguation was necessary. Our disambiguation builds on a subsequent reduction of candidate UniProt IDs for each recognized name (see our GN task paper.) The highest impact on performance came with the reduction to organisms. We used the Ali Baba tool to recognize organism names in the corresponding abstracts. We compared these identified organisms to the annotations of each potential UniProt entry. Comparison was based on the controlled vocabulary provided by UniProt [6], which we enriched using the NCBI Taxonomy to include other common names, as well as manual curation (so that “patients” would trigger “human.”) Sometimes it was not possible to restrict the IDs to only one candidate. In such cases, we would report the first standard name (for higher precision) or all remaining (higher recall.) We noticed that in most cases, at least one standard name (out of a predicted PPI pair) was correctly found by the disambiguation. When the second was not correctly found, however, this still accounted for an overall false positive and a missing annotation.

Interaction extraction

We applied a sentence alignment against a pre-compiled set of patterns on every sentence that contained at least two proteins. Such patterns describe typical occurrences of evidences that mention PPIs. Very simple examples for these would be [*protein* binds to *protein*] or [*protein* bound to the *word* domain of *protein*]. Here, *protein* and *word* are wildcards for every protein recognized by the first component, or arbitrary words, respectively. To find such patterns, we applied the following strategy. First, we collected a large set of sentences from PubMed that most likely describe PPIs. To find such sentences, we used the IntAct database [5] and searched for sentences that contain an interaction pair in PubMed. Each protein in IntAct can be mapped to a UniProt ID and, using the above recognition, we scanned the full PubMed database for any occurrence of a pair of proteins known to interact. We reduced each sentence to the core phrase (potentially) describing the interaction and searched for typical words (“binds”, “associated”, “complex”, etc.) For more details, please refer to [4], examples are shown in Table 1. Starting with a set of more than 200,000 such sentences, we computed a pairwise similarity using sentence alignment. The input for these alignments were tokens, token stems, and part-of-speech tags for each position in a sentence. A distance matrix containing pairwise alignment scores for all pairs of core phrases was used to construct a guide tree for clustering (comparable to ClustalW.) On each cluster, we then performed a multiple sentence alignment to compute a consensus pattern that best describes the sentences. Figure 1 shows an example for an MSA to compute such a consensus pattern (POS tags only.) We found ca. 10,000 such consensus patterns, many of which were as simple as the aforementioned examples, but with many rather complex patterns as well.

Sentence alignment provides an inexact matching strategy for sequences of words; this allows for (often observed) deletions or insertions of words with minor influence on the overall statement (for instance, adjectives and determiners.) Consensus patterns bring two main advantages: i) they consist

of word sequences actually observed in evidence texts and are thus very specific; ii) they combine observations made across multiple evidences into one pattern and thus generalize well.

protein binds to *protein*
(protein) binds to its **receptor** (*protein*)
protein binds to the cytoplasmic tail of *protein*
protein recruits the adapter molecule *protein*
protein site was specifically recognized by c- *protein*
protein and protein compete for binding to *protein*
(protein) results in decreased protein synthesis
Arabidopsis *protein* (*protein*) **associates** with both *protein* and *protein*
cytosolic *protein* is **associated** with a **complex** of *protein* (*protein*)
protein, a modular **adapter** which in muscle cells **interacts** with members of the *protein* family including *protein*
protein induces activation of coagulation and fibrinolysis through an exclusive **effect** on the *protein*
protein was previously found to **interact** with the KRAB silencing domain of *protein* and with the *protein*

Table 1: Examples for phrases collected from PubMed. Sentences were reduced to their core. *protein* indicates proteins of arbitrary name, while all other words and symbols appeared as such; interaction words are **bold**.

Analysis

Short description	Precision	Recall	F1 (in%)
<i>min=2; ids=2; organism=a,h,m,y,1</i>	7.7	69.4	13.2
<i>min=3; ids=1; organism=a,1</i>	15.0	65.1	22.5
<i>min=1; ids=1; organism=a,1</i>	44.5	41.7	40.5

Table 2: Results for different strategies on the IPS test set. *min*, minimum of identified interactions per pair and article required for a prediction; *ids*, number of submitted IDs per protein in case more than one was left after NEN; *organism*, order of assignment to organisms for unresolved proteins: take organism found in abstract, take human, mouse, yeast, or highest ranked gene (1).

Table 2 shows the results of our method depending on different settings. First, we see that proper NEN was crucial regarding the overall outcome. We found that associating a protein with an organism was quite easy, and our paper for the GN task discusses how intra-organism ambiguities could be solved. We encountered most NEN-related problems as a result from erroneous PDF to text conversion, an issue that has been discussed elsewhere. For example, in many of the plain texts, Greek letters, which were crucial to identify the right member of a family, were missing. Some false positive predictions were found as discussed in dangling text or not annotated in the gold standard for various reasons (different understanding of an interaction; not main thrust of publication.) Thus, tuning towards IPS-task-specific annotations on the training corpus might help. Regarding the “main thrust”, we found that many of the PPIs were discussed quite often within a single publication, so even requiring at least three evidences did not influence the recall much, but increased the precision. PPIs mentioned only once in the Introduction, for instance, could be filtered out. Evaluations of our approach on other corpora revealed quite different results. On the SPIES corpus [2], the method showed a precision around 80% at 50% recall. There are two differences compared to the results on IPS: (1) the figures are lower in general and (2), the order of precision and recall have changed. NER/NEN is not necessary for SPIES, which consists of 1000 single sentences that all contain at least one PPI.

References

- [1] Blaschke, C. and Valencia, A., The Frame-Based Module of the SUISEKI Information Extraction System, *IEEE Intelligent Systems*, 17(2):14–20, 2002.
 - [2] Huang, M., Zhu, X., Hao, Y., Payan, D.G., Qu, K., and Li, M., Discovering patterns to extract protein-protein interactions from full texts, *Bioinformatics*, 20(18):3604–3612, 2004.
 - [3] Plake, C., Schiemann, T., Pankalla, M., Hakenberg, J., and Leser, U., AliBaba: PubMed as a graph, *Bioinformatics*, 22(19):2444–2445, 2006.
 - [4] Hakenberg, J., Leser, U., Kirsch, H., Rebholz-Schuhmann, D., Collecting a large corpus from all of Medline, *Proc. Symposium on Semantic Mining in Biomedicine*, 2006.
 - [5] See <http://www.ebi.ac.uk/intact/>
 - [6] See <http://www.expasy.org/cgi-bin/speclist>



Identifying Protein-Protein interactions in Biomedical publications

Alejandro Figueira Günter Neumann
figueroa@dfki.de neumann@dfki.de

DFKI - LT Lab, Stuhlsatzenhausweg 3, D - 66123, Saarbrücken, Germany

Abstract

The paper describes the approaches and the results of our participation in the protein-protein interaction (PPI) extraction task (sub-tasks 1 to 3) of the BioCreative II challenge.¹ The core of our approach is to analyse the logical forms of those sentences which contain the mentioning of relevant protein names, and to rank the sentences from which the relations were extracted using the class descriptors computed in the sub-task 1 and interaction sentences from the Christine Brun corpus.

Keywords: Protein-Protein interactions identification, Predicate Analysis

1 Introduction

One of the goals of the Question Answering group at the DFKI LT-Lab is taking part in standard evaluations such as TREC or CLEF. During the last three years, our group has focused on the Cross-Lingual German-English, English-German and monolingual German tracks of the CLEF campaign. Results have been strongly encouraging, obtaining the best results for these tracks [13, 14, 15].

In QA the current research focus is still on domain-open QA in order to answer term-based questions like *Where was the “killer smog” of 1952 which resulted in 4,000 deaths?* from newspaper articles. However, there is an increasing interest to explore also domain-specific QA, i.e., to answer domain-specific questions from domain-specific sources. Here, event specific questions are of interest, which require the identification of relevant relation instances, e.g., in order to answer a question like *How does GUKH interacts with DLG?* from scientific articles.

Our approach is to consider domain-specific QA as a kind of *on-demand information extraction* where the NL question describes important constraints for the relation instances that have to be extracted from the answer sources. This perspective actually motivated our interest in the BioCreative challenge, especially in the Protein-Protein interaction subtask. Of course, the focus in the BioCreative challenge is on off-line information extraction in the sense that the information request (i.e., the question) is pre-specified and that all possible valid relation instances have to be extracted (i.e., the answer candidates). For researchers in question answering like us, there are important subtasks in common for on-demand and off-line information extraction, like named entity recognition, relation mining, co-reference detection, concept name disambiguation, etc.

Since BioCreative II was our first excursion into Information Extraction in Biology, our objectives were: (a) learn about the inherent challenges and share our experience, and (b) discriminate key components of systems that deal with natural language texts in the biological domain. Here, the main motivation raises from the way that biological texts are written: a plenty of technical words and

¹The work presented here was partially supported by a research grant from the German Federal Ministry of Education, Science, Research and Technology (BMBF) to the DFKI project HyLaP (FKZ: 01 IW F02) and the EC-funded project QALL-ME.

complex sentence structures as well as a high term variation, especially gene names. Assessing several Natural Language Processing techniques is hence positively encouraging, and by the same token, our group focused essentially on covering the sub-tasks (a) Protein Interaction Article Sub-task (IAS), (b) Protein Interaction Pairs Sub-task (IPS), and (c) Protein Interaction Sentences Sub-task (ISS).

In the next section, we firstly describe our principle approach and then focus on particular solutions for the different sub-tasks in the subsequent sections. In section 6 we briefly discuss our results, which – of course – we interpret as “the glass is half full”.

2 Predicate Analysis

Predication computes the semantic representation of a sentence. This representation distinguishes relationships or semantic roles played by its different constituents within a semantic frame[10]. To neatly illustrate this, consider the sentence “*GUKH interacts with DLG in vivo*”, its corresponding predicate representation is given by:

interact(“GUKH”, “with DLG”, “in Vivo”)

In this representation, the verb is the predicate and the remaining constituents are arguments. Labels are then assigned to each argument according to their role in the predicate. The level of specification can be abstract such as VERB, SUBJECT, OBJECT, or specific to the different framesets of a particular verb. Good examples are the two framesets for the verb “*inhibit*” (see [11] for examples in the PropBank[16]):

1. **inhibit**(preventor entity, thing prevented from happening), i. e. “*Influenza virus NS1 protein inhibits pre-mRNA splicing*”.
2. **inhibit**(preventor entity, thing prevented from happening, medium), for instance: “*ArhGAP9 inhibits Erk and p38 activation through WW domain binding Boon K Ang1 ,2*”.

Each frameset is seen as a different semantic frame. The motivation behind applying predication to discriminate protein interaction is two-fold: (a) since proteins interactions are likely to be expressed by complex semantic constructions at the sentence level [4, 6], and (b) the existence of tools, like MontyLingua[17], which compute a semantic representation of a raw text in English. MontyLingua specifically extracts tuples *verb(subject, objects)*, which are an abstract predicate-argument representation of sentences in a given text.

3 Document Classification

In this sub-task, documents containing relevant protein interaction information must be accurately identified. This identification must be performed by accounting solely for their headlines and abstracts. For this purpose, systems were allowed to submit three different runs, and in our case, to test three different strategies. Two out of these three strategies started stepwisely pre-processing the training and testing sets as follows:

1. **Protein name removal** Since protein and gene names are the most obvious source of classification bias[1], they are distinguished by Abner[18] and replaced with the word “*Protein*” afterwards.
2. **Lemmatization** In this step, words are lemmatized by means of MontyLingua[17], in order to avoid counting several morphological inflections of the same term as occurrence of different words.

3. **Sentence normalization** Abstracts are split into sentences by means of JavaRap[19] and normalized afterwards. This normalization consists chiefly in inserting spaces between punctuation and words, this way our methods avoid also misinterpreting words followed by their punctuation as occurrence of different words. By the same token, all words are lowercased.
 4. **Bag of words** Each abstract is represented as a bag of words. These words are distinguished by means of spaces and every word is linked to their frequency on the corresponding abstract. Stop-words² are removed from each bag.

While our strategies were dealing with this task, we found that the unbalanced training data, caused by the strong bias in favour of positive samples, was a major problem. Consequently, strategies aiming specifically for dealing with unbalanced data were explored. The first two runs (**RUN I** and **RUN II**) were based on the binary Bayes classifier presented in [2]. In these runs, we trained two classifiers: one with abstracts and the other with headlines. Documents in the test set were eventually ranked by weighting the output of both classifiers in the following way:

$$r_d(D) = \begin{cases} r_h(D) * r_a(D) & \text{if } r_h(D) <> 0 \text{ and } r_a(D) <> 0. \\ r_h(D) & \text{if } r_a(D) == 0. \\ r_a(D) & \text{if } r_h(D) == 0. \end{cases}$$

Where $r_h(D)$ and $r_a(D)$ are the output (corresponding to a document D) of the Maximum Entropy classifier trained with headlines and abstracts respectively. A new document D was considered containing relevant protein interaction information, if $r_d(D) > 1$, otherwise irrelevant. The training tuples were chosen by means of a 10-fold validation and due to three reasons, they were deliberately selected only from negatives and noisy positives samples: (a) we found that positive samples did not improve results, (b) markedly reduce the size of the training set, and (c) given the fact that the test set belongs solely to the positive and negative class, we clearly intended to increase the robustness of our classifiers by decreasing their dependence upon positive samples. These first two runs differ fundamentally in the training model obtained by the 10-fold cross validation.

RUN III was based on the approach presented in [9]. In this approach, documents and categories are seen as sets of independent words. For each category, this classifier creates two data structures: semantics-oriented topic words and surface focused index words with a high discrimination value. Documents are classified by means of two category rankings (each for index and topic words) which are combined to one ranking (m-ary classifier) afterwards. This classifier was trained with non pre-processed negative and positives samples only.

4 Protein protein interaction identification

This sub-task aims at recognising protein interactions from full text articles. The underlying assumption of our methods is that interacting proteins are expected to co-occur in many sentences along the respective article, and therefore, in several semantic frames. Some of these semantic frames are accordingly more likely to indicate whether they interact or not. The flow of our strategy is as follows:

1. **Pre-processing** starts by extracting the content from the PDF2TXT version of the article and splitting it into sentences by means of JavaRap[19] afterwards. The higher frequent sentence was interpreted as the title or headline of the article, since it is seldom directly recognised from the text and it is usually repeated. Like [3], citations were permanently removed by means of purpose-built regular expressions, this way the quality of the predicate analysis noticeably improves. Another key issue is that sections within documents are identified by searching for special

²The stop-list from [20] is used. It contains 319 highly frequent closed class forms.

tags such as “*MATERIALS*”, “*REFERENCES*”, “*ACKNOWLEDGMENTS*”. In case that no section was correctly identified, the article is seen as containing only one section. Sentences are then associated with their corresponding sections afterwards.

2. **Protein detection** is performed by Abner across the whole document. Since our system works with predicates at the sentence level, protein references across sentences must be unveiled. For this specific purpose, we took advantage of the full implementation of [5] provided by JavaRap, instead of its partial implementation presented in [4].
3. **Predicate Analysis** takes all sentences containing at least two recognised proteins and identifies its predicate and arguments. This semantic structure is a crucial aspect of our strategy (also in [4, 6]), because the role of proteins within sentences signals their relation and verbs whether this relation a protein-protein interaction is or not [4, 3, 7]. Arguments with no protein mentions were for this reason also completely discarded. Another thing is, headlines of articles are usually ungrammatical, MontyLingua could not then distinguish their structure. Our system keeps hence track of co-occurring proteins within headlines, because they are likely to signal a relevant relation.
4. **Gene name normalisation** maps protein names, which occur in at least one predicate, to their corresponding **UniProt Accession Numbers**. This mapping consists of the next steps:
 - (a) The **UniProt light Knowledge Base** was indexed by normalized terms extracted from the following columns: description and gene name lines, gene synonyms, locus and ORF names, keywords. These terms indexed their corresponding accession numbers and their normalization consisted in leaving only letters and numbers [8].
 - (b) Candidate protein keys are extracted by looking for matches across this index. Firstly, our system attempts to find exact string matches, if it does not succeed, it looks for inexact matches. The first matching considers only the exact gene name identified in the text, and the second accounts solely for the letters and number in the distinguished gene name.
 - (c) Our system searches for co-occurring pairs organism-protein within sentences. If any highly co-occurring pair exists, the organism is used for disambiguating the key.
 - (d) If key ambiguity still exists, our system tries to discover known interacting key pairs in the ExPasy Knowledge Base[21].
 - (e) If our system cannot disambiguate the key, the first key in alphabetical order is selected.

Protein names were eventually replaced in predicates with their mapped accession numbers. Each predicate provided accordingly the following interacting pairs:

- (a) The subject was paired with each argument.
- (b) Each argument was paired with the other arguments.
5. **Ranking predicates and protein pairs** Let S be the set of $1 \leq s \leq |S|$ sentences extracted from a given article D and S_s the s -th sentence in S , $1 \leq s \leq |S|$. Each sentence $S_s \in S$ is then ranked according to the potential of its words for expressing protein interactions. The computation of this potential is based mainly on the following equation:

$$\text{word_potential}(S_s) = \sum_{\forall w_i \in S} P^{ISS}(w_i) + W^{IAS}(w_i)$$

Where $P^{ISS}(w_i)$ is the probability that the word w_i occurs within interaction sentences across abstracts in the Christine Brun corpus. $W^{IAS}(w_i)$ is given by:

$$W^{IAS}(w_i) = W^+(w_i) - W^-(w_i)$$

Where $W^+(w_i)$ and $W^-(w_i)$ are the likelihood of w_i to the noisy positive and negative class respectively (previously computed in sub-task I (see section 3)). Additionally, we define the potential of a verb for expressing protein interactions as P_{verb}^{IAS} , the probability that a protein and a particular verb co-occur in the same sentence across positive and noisy positive abstracts given in sub-task I. The rank of a sentence is eventually defined as follows:

$$rank(S_s) = \Gamma * (1 + word_potential(S_s)) * (1 + \sum_{\forall \vartheta_r \in \vartheta(S_s)} P_{verb}^{IAS}(verb(\vartheta_r))) \quad (1)$$

Where $\text{verb}(\vartheta_r)$ is a function which returns the verb in the predicate ϑ_r , $\vartheta(S_s)$ a function which returns the identified predicates for S_s , and Γ is a weight according to the section in which S_s occurs. $\Gamma = 1$ for all sections, apart from “*MATERIALS*”, “*MATERIALS AND METHODS*”, “*RESULTS AND DISCUSSION*”, “*RESULTS*”, “*EXPERIMENTAL*”, “*DISCUSSION*”, “*EXPERIMENTAL PROCEDURES*”, which their value for Γ was set to two. The rank of the interaction of two proteins p_1 and p_2 is given by:

$$rank(p_1, p_2) = \tau(g_1, g_2) \gamma \sum_{\forall S_s \in S} \lambda(p_1, p_2, S_s) * rank(S_s)$$

Where $\lambda(p_1, p_2, S_s)$ is the number of predicates $\vartheta_r \in \vartheta(S_s)$ in which p_1 and p_2 occur. The weight γ favours pairs occurring in the title. $\tau(g_1, g_2)$ favours interaction pairs that can be found in the Expasy Knowledge Base (step 4.d).

6. The three runs were generated according to the following criteria:

 - (a) **RUN I:** All identified ranked pairs.
 - (b) **RUN II:** All ranked pairs that satisfactorily fulfil the next rule

$$rank(p_1, p_2) > 0.1 * rank^*$$

Where $rank^*$ is the rank value of the higher ranked pair.

- (c) **RUN III:** Top five ranked pairs.

5 Protein protein interaction sentence Ranking

This sub-task asks participants to provide, for each protein interaction pair, a ranked list of at most five text passages (maximal three sentences per passage) describing their interaction. For this sub-task, we submitted only one run. Our system took advantage of the ranking provided by sub-task II (eq. 1) and selected the top five ranked sentences for each protein interaction pair. Each sentence was aligned with the source HTML document as follows:

1. The first word in the sentence was used as an anchor. This anchor signals the start of a window of two times the length of the ranked sentence.
 2. Words were placed in each window according to their relative position within the ranked sentence. When a word could not be accurately located within the window, it was marked with a “*”. The window with less “*” was eventually selected.
 3. If the last word in the selected sentence was properly aligned, the window is cut off at the end of this word.

6 Results

The following section describes the results obtained by our system in details.

6.1 Document Classification

Table 1 and 2 provide the results obtained by each run for the document classification sub-task:

	Table 1: Results overview.					
	Precision	Recall	Accuracy	F-Score	AUC	Error Rate
RUN I	0.527	0.986	0.550	0.687	0.795	0.44
RUN II	0.518	0.992	0.536	0.681	0.797	0.46
RUN III	0.577	0.725	0.597	0.643	0.589	0.40

RUN I and RUN II finished with a F-score about the mean of all systems (0.6868). Conversely, RUN III achieved a slightly worse F-score, but a higher accuracy. Table 2 shows the confusion matrices for each run:

	Table 2: Confusion matrices.			
	TP	FP	TN	FN
RUN I	370	332	43	5
RUN II	372	345	30	3
RUN III	272	199	176	103

The number of FP gives the reason for the high recall and low precision of RUN I and RUN II, caused by the assignment of many negative test documents to the positive class. Table 2 also shows that RUN III improved the recall of the negative class at expenses of its precision, which is a consequence of the few number of negative training samples used for our classifiers. Table 3 provides greater details about the results achieved by the three runs:

	Table 3: Comparisson of the three runs.		
	RUN I	RUN II	RUN III
RUN I	-	19	116
RUN II	6	-	109
RUN III	153	158	-

This table compares two runs by taking documents, for which their prediction differ, and counting the number of correct forecast for each run. For instance, RUN I and RUN II obtained different predictions for 25 documents and six cases were correctly labelled by RUN II, while 19 cases by RUN I. This result envisages that the combination of the output of several classifiers can improve results.

6.2 Protein-protein interaction identification

Protein-protein interaction prediction

Tables 4 and 5 supply our per document and overall results respectively. In these tables, EVAL stands for all articles and SP_EVAL for the subset containing exclusively SwissProt interaction pairs.

The total recall of our system was about the mean respecting the 45 runs submitted by all systems. In case of EVAL, our system achieved 0.09 (0.1064 overall) and in case of SP_EVAL, it finished with 0.094 (0.1150 overall). In contrast to recall, results concerning precision are unconvincing. Given this sharp difference, it can be concluded that our system discovers interacting pairs of proteins along with a large amount of incorrect pairs. Looking closer upon table 5, we additionally observe that the decrease in recall from RUN I to RUN II and RUN III leads us to conclude that interaction pairs tend

Table 4: Mean values for the three different runs (per document).

	EVAL			SP_EVAL		
	Precision	Recall	F-Score	Precision	Recall	F-Score
RUN I	0.01	0.11	0.018	0.011	0.11	0.019
RUN II	0.029	0.056	0.035	0.025	0.056	0.032
RUN III	0.026	0.087	0.036	0.023	0.087	0.034

Table 5: Overall result for the three different runs.

	EVAL			SP_EVAL		
	Precision	Recall	F-Score	Precision	Recall	F-Score
RUN I	0.01	0.09	0.018	0.01	0.094	0.019
RUN II	0.029	0.030	0.034	0.025	0.026	0.026
RUN III	0.018	0.05	0.027	0.019	0.05	0.027

to be ranked low (RUN II and RUN III consider only a subset of the highest ranked pairs of RUN I). These conclusions motivate the usage of Montylingua for distinguishing protein interactions, but a strategy that can filter out misleading interactions along with a better ranking strategy is necessary, this way the noise could be reduced and the precision similarly increased.

Interactor proteins Normalisation.

Tables 6, 7 and 8 gives our results for the normalisation of interactors.

Table 6: Mean values for interactor proteins normalization (all evaluated articles).

	EVAL			SP_EVAL		
	Precision	Recall	F-Score	Precision	Recall	F-Score
RUN I	0.06	0.29	0.095	0.066	0.32	0.11
RUN II	0.11	0.18	0.13	0.11	0.19	0.135
RUN III	0.09	0.20	0.11	0.095	0.22	0.123

Table 7: Mean values for interactor proteins normalization (all evaluated articles with predictions).

	EVAL			SP_EVAL		
	Precision	Recall	F-Score	Precision	Recall	F-Score
RUN I	0.06	0.31	0.1	0.072	0.34	0.11
RUN II	0.14	0.23	0.17	0.15	0.26	0.18
RUN III	0.11	0.27	0.15	0.13	0.30	0.17

Our gene normalisation strategy achieves a slightly better recall than the mean considering all evaluated documents and a slightly worse recall taking into account only articles with predictions. In the three cases, RUN II was the best, because of its higher precision and F-Score. The higher recall of RUN I is a logical consequence of accounting for an unfiltered set of pairs.

Table 9 provides the performance of our gene mormalisation strategy: 361 out of 1306 protein names were correctly identified and correctly mapped to their database entries, and 268 out of 896 taking into account only SwissProt entries. The difference in the number of correctly identified protein names shows that our ranking strategy ranks many relevant interacting proteins low. This could be

Table 8: Mean values for interactor proteins normalization (Overall SwissProt interactor pairs).

	EVAL			SP_EVAL		
	Precision	Recall	F-Score	Precision	Recall	F-Score
RUN I	0.06	0.28	0.09	0.04	0.3	0.074
RUN II	0.13	0.15	0.14	0.097	0.158	0.12
RUN III	0.09	0.18	0.12	0.064	0.19	0.096

Table 9: Number of interactor protein-article associations.

	EVAL				SP_EVAL			
	Correct	Wrong	Missed	Predicted	Correct	Wrong	Missed	Predicted
RUN I	361	6011	945	6372	268	6104	628	6372
RUN II	197	1273	1109	1470	142	1328	754	1470
RUN III	238	2421	1068	2659	171	2488	725	2659

due to the detection of sentences, some relevant sentences could not be parsed, therefore, the relation between proteins could not be properly determined. Results show that this is the most critical module in our system.

6.3 Protein-protein interaction sentence ranking

Our system found out 590 sentences that matched the gold standard (manually selected passages), 285 out of these 590 were unique. Since our system returned a long list of interacting proteins in sub-task II, it returned a huge list of 21431 sentences for this sub-task (10422 unique), which caused an MMR of 0.3785.

7 Conclusions

In this work, we presented our first participation in an evaluation of Information Extraction Systems in Biology. For a future participation, we envisage the following improvements:

1. Combining the output of several classifiers in order to enhance the accuracy of our predictions and the robustness of our classifier.
2. The usage of language models that consider more contextual information, like bi-grams.
3. A bootstrapping strategy can also take advantage of recognised pairs, this way undetected sentences by Montylingua can be identified, bringing about an improvement in the ranking of sentences and interacting protein pairs.
4. The usage of LSA[12] and the Web for discriminating the source organism of a protein.

References

- [1] Marcotte, E. M., Xenarios I. and Eisenberg, D, *Mining literature for protein-protein interactions*, Bioinformatics, 17:4, pp. 359–363, 2001.
- [2] Rennie, J. D. M., Shih, L., Teevan, J. and Karger, D. R., *Tackling the Poor Assumptions of Naive Bayes Text Classifiers*, in Proceedings of ICML-2003, Washington DC, 2003.

- [3] Huang, M., Zhu, X., Hao, Y., Payan, D. G., Qu, K. and Li, M., *Discovering patterns to extract protein-protein interactions from full texts*, Bioinformatics, 20:18, pp. 3604–3612, 2004.
 - [4] Sekimizu, T., Park, H. and Tsujii, J, *Identifying the interaction between Genes and Gene Products Based on Frequently Seen Verbs in Medline Abstracts*, In Genome Informatics Series: Proceedings of the Workshop on Genome Informatics, Vol. 9, pp. 62–71, 1998.
 - [5] Lappin, S. and Leass, H. J., *An algorithm for pronominal anaphora resolution*, Computational Linguistics, 20:4, pp. 535–561, 1994.
 - [6] Ahmed, S., Chidambaram, D., Davulcu H. and Baral C., *IntEx: A Syntactic Role Driven Protein-Protein Interaction extractor for Bio-Medical Text*, in Proceedings ACL-05/ISMB-05, pp. 54–61, 2005.
 - [7] Hatzivassiloglou V. and Weng W., *Learning Anchor Verbs for Biological Interaction Patterns from Published Text Articles*, Int J Med Inf., 67, pp. 19–32, 2002.
 - [8] Wellner, B., *Weakly Supervised Learning Methods for Improving the Quality of Gene Name Normalization Data*, in Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, pp. 1–8, Detroit, June, 2005.
 - [9] Neumann G. and Kappes M., *A simple base-line text-categorizer for evaluating the effect of feature extraction in text mining applications*, in abstract booklet accompanying the 26th Annual Conference of the German Classification Society (GfKI 2002), July 22-24, 2002, University of Mannheim, Germany.
 - [10] Gildea D. and Jurafsky D., *Automatic Labeling of Semantic Roles*. Computational Linguistics, Computational Linguistics, 28:3, pages 245–288, 2002.
 - [11] Palmer, M., Gildea D. and Kingsbury P., *The Proposition Bank: An Annotated Corpus of Semantic Roles*, Computational Linguistics, 31:1, pages 71–106, 2005.
 - [12] Deerwester, S., Dumais, S., T., Furnas, G., W., Landauer, T., K. and Harshman R., *Indexing By Latent Semantic Analysis*, Journal of the American Society For Information Science, 41, 391–407, 1990.
 - [13] Sacaleanu B. and Neumann G. *DFKI-LT at the CLEF 2006 Multiple Language Question Answering Track*. In Working Notes for the CLEF 2006 Workshop, August, Alicante, Spain, 2006
 - [14] Neumann G. and Sacaleanu B. *DFKI's LT-lab at the CLEF 2005 Multiple Language Question Answering Track*. In Working Notes for the CLEF 2005 Workshop, 21-23 September, Vienna, Austria, 2005.
 - [15] Neumann G. and Sacaleanu B. *Experiments on Robust NL Question Interpretation and Multi-layered Document Annotation for a Cross-Language Question/Answering System*. In: C. Peters et al. (Eds): Clef 2004, LNCS 3491, pp. 411-422, 2005, Springer Berlin Heidelberg.
 - [16] <http://www.cs.rochester.edu/~gildea/PropBank/Sort/>
 - [17] <http://web.media.mit.edu/~hugo/montylingua/>
 - [18] <http://www.cs.wisc.edu/~bsettles/abner/>
 - [19] <http://www.comp.nus.edu.sg/~qiul/NLPTools/JavaRAP.html>
 - [20] http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words
 - [21] <http://www.expasy.org/sprot/>



Integrating knowledge extracted from biomedical literature: normalization and evidence statements for interactions

Graciela Gonzalez¹
`graciela.gonzalez@asu.edu`

Luis Tari²
`luis.tari@asu.edu`

Anthony Gitter²
`anthony.gitter@asu.edu`

Robert Leaman²
`bob.leaman@asu.edu`

Shawn Nikkila²
`shawn.nikkila@asu.edu`

Ryan Wendt²
`ryan.wendt@asu.edu`

Amanda Zeigler²
`amanda.zeigler@asu.edu`

Chitta Baral²
`chitta@asu.edu`

¹ Department of Biomedical Informatics, ² Department of Computer Science and Eng, School of Computing and Informatics, Fulton School of Engineering, Arizona State University, Tempe, AZ 85281 USA,

Abstract

This paper reports our approach to three specific tasks of the BioCreAtIVe II challenge: protein interaction sentences (PPI-ISS), protein interaction pairs (PPI-IPS) and gene normalization (GN). Our approach to software engineering and implementation decisions was based on addressing first and foremost the core problem of integrating knowledge extracted from the literature: thus, we saw PPI-ISS as pairing statements of certain characteristics with core facts extracted elsewhere in the document and GN as mapping extracted entities to some standard names. This allows us to focus on generic solutions that can then be gradually refined to solving specific problems. In this same spirit, we developed a text-extraction XML format, a query language for the extraction of information constructs from a parse tree, a prototype extraction system, and a prototype web-based generic evaluation system that were then adapted to BioCreAtIVe. Our approach to the three tasks as well as analysis of results and a brief description of the related technologies developed are included in this report.

Keywords: normalization, protein-protein extraction, NLP, ranking, evaluation, data mining

1 Introduction

Numerous efforts to extract and annotate data from biomedical articles have resulted in over 200 databases and other resources [1] that allow scientists to access (in most cases, free of charge) structured biological information. However, it is estimated that between 300,000 and 500,000 [2] articles are added each year to the millions already in PubMed. The constantly increasing number of articles and the complexity inherent to its annotation results in data sources that are continuously outdated. For example, GeneRIF (Gene Reference Into Function), was started in 2002, yet it covers only about 1.7% of all the genes in Entrez [3] and 25% of human genes.

Automatic extraction and annotation seems a natural way to overcome the limitations of manual curation, and a lot of work has been done in this area, including the automatic extraction of genes and gene products [4], protein-protein interactions [5-9], relationships between genes or proteins and biological functions[10], genes and diseases[11-13], and genes and drugs[14], among others. However, the reliability of the extracted information varies greatly, and thus discourages the biologists from using it for their research.

The BioCreAtIVe II challenge with its different tasks addresses core areas in automatic extraction from biological texts: gene mention, gene normalization, and protein-protein interaction extraction. A particularly challenging aspect of the later is that only interactions that were supported by evidence of experimental methods in the same article were of interest¹. The KDD Cup 2002 Information Extraction challenge [15] was

¹ Quoting from the 1st paragraph of the IPS Evaluation Process readme file, "... interaction pairs were only annotated by the database curators from the full text articles of the test set in case there was an experimental confirmation for this interaction mentioned in the article."

among the first to propose extracting interactions accompanied by sentences describing the experimental evidences. The logic behind this requirement is very important and often overlooked by PPI extraction systems: in practice, only interactions which are confirmed using experimental techniques are useful for high quality interaction annotations for biologists, and such sentences are often used by human curators as a deciding factor when annotating protein-protein interactions from text. Two of the most important manually annotated PPI databases, IntAct [16] and MINT [17], use this criteria and include the sentences in their databases. Usually, automated interaction extraction systems [5-9, 18] deploy techniques to determine if sentences are about interactions, but do not particularly address the more semantically refined concept of whether the given sentences provide *evidence* of the interaction. We hypothesize that the disparity in performance of the systems participating in BioCreAtIVe with respect to what is reported in the literature for PPI extraction systems (for example, reaching 92% f-measure in [18]) can be attributed in part to this requirement, as well as to the fact that such reported performance measures might in reality not be comparable, given the disparity in evaluation methods and gold standards used to generate them.

This paper reports our approach to three specific tasks of the BioCreAtIVe II challenge: interaction support statements (PPI-ISS), protein-protein interaction extraction (PPI-IPS) and gene normalization (GN) that share a number of pre and post processing techniques. Our approach to software engineering and implementation decisions was based on addressing the core problem of integrating knowledge extracted from the literature: thus, we saw PPI-ISS as pairing statements of certain characteristics to core facts extracted elsewhere in the document and GN as mapping extracted entities to some standard names. This allows us to focus on generic solutions that can then be refined to specific problems. Such refinements include, for example, the use of specific ontologies (like the MeSH category "Investigative Techniques" for locating evidence statements) and filtering and ranking techniques (like those applied to extracted interactions to find the most likely true positives).

2 Method and Results

2.1 Protein Interaction Sentences (PPI-ISS)

In this section, we describe our approach for the PPI-ISS task to extract passages that contain experimental confirmation for extracted. The system takes as input extracted protein-protein interactions and their corresponding PubMed ids, and outputs a ranked list of passages which describe the experimental evidence for the interactions. As part of the requirement of the PPI-ISS task, a maximum of 5 passages per interaction is returned and each passage cannot be longer than 3 sentences.

2.1.1 PPI-ISS Architecture

The system architecture for passage extraction is illustrated in Figure 1. The system uses Lucene [19] to index an XML version of the articles, which are converted in-house from the BioCreAtIVe HTML corpus. For each interaction extracted by our extraction systems (described in Section 2.2), a query is formed to retrieve potentially relevant paragraphs from the corresponding article. Passages are then extracted from within the relevant paragraphs. Each passage is scored based on the proteins and experimental methods they contain, to produce a final ranked list of passages. The details of each of the major components follows.

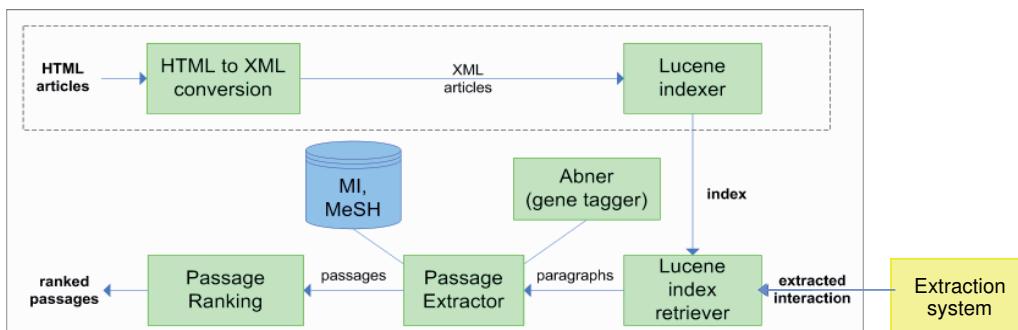


Figure 1. System architecture for extracting passages with experimental evidences. All articles are pre-processed by converting them to XML and indexing the resulting files in Lucene. Given an interaction pair, a Lucene query results in relevant paragraphs from the corresponding article. The extraction systems used are considered separately (Section 2.2).

Retrieval of relevant paragraphs. We used the BioCreAtIvE HTML corpus as our initial input, converting it into an XML format. Each paragraph is stored in the XML format as an element of generic sections which include abstract, introduction, methods, results, conclusion, references and captions. Note that not all articles explicitly title their sections as such, so the mapping of paragraphs to sections is done using a heuristic algorithm. The XML file moves through the different system components as a universal input/output format, since all relevant information is added to it. For example, the acronym resolution algorithm described in [20] is run on the whole article, and occurrences of the acronyms are stored as elements in the XML file.

The XML articles are indexed using Lucene [19]. Given an interaction pair, a Lucene query returns paragraphs that have mentions of both of the entities in the interaction, and the section to which they belong. All relevant paragraphs are processed to extract valid passages, as detailed next.

Extraction of passages. The passage extraction component takes an interaction of interest and the relevant paragraphs as input, and produces a ranked list of passages as output. A *passage* is defined as a contiguous list of up to 3 sentences. To find passages, the sentences in the relevant paragraph are scanned and its genes and proteins are tagged using ABNER [21] (trained based on BioCreAtIVe I corpus). A sentence with one or both of the interactors serves as *seed* for a passage. If relevant keywords are found in the neighboring sentences, they are added to the passage. Keywords of interest include the protein interactors and terms associated with experimental evidence.

To recognize experimental methods within a passage, a dictionary of stemmed experimental method terms was compiled from the Molecular Interaction ontology (MI) [22] and MeSH terms under the categories “Investigative Techniques”, “Diagnosis” and “Therapeutics”. In each of the sentences in the passages, words are stemmed using the Porter stemmer [23] and exact string-matching is used in for recognizing them.

A passage is *valid* if it includes both of the proteins in the interaction. Valid passages are scored based on two criteria: (1) origin of the passages, and (2) frequency of terms of interest. The intuitive basis for criteria (1) is that experimental evidence for protein-protein interactions is usually mentioned in the methods and/or results sections as well as in captions more often than in other sections. Thus, a passage p_i that originated from one of these sections is scored higher, as follows:

$$score_origin(p_i) = \begin{cases} 1 & \text{if } p_i \text{ is originated from method, results, captions of an article} \\ 0.5 & \text{if } p_i \text{ is originated from abstract, introduction, conclusion of an article} \\ 0 & \text{if } p_i \text{ is originated from the references section of an article} \end{cases}$$

Criteria (2) is based on the number of experimental methods and gene/protein names of interest appearing in the passages. Let $freq(p_i)$ be the number of occurrences of experimental methods and gene/protein names of interest (interactors and their synonyms) in passage p_i , where p_1, \dots, p_n are valid passages extracted from an article. Let $F = \{ freq(p_1), \dots, freq(p_n) \}$. Then criteria (2) is computed as follows:

$$score_evidence(p_i) = freq(p_i) / \max F$$

The final score of passage p_i is the sum of $score_origin(p_i)$ and $score_evidence(p_i)$. This single score is associated with each valid passage. The top 5 passages from all relevant paragraphs are returned.

2.1.2 PPI-ISS Analysis

We submitted 3 runs for the BioCreAtIVe PPI-ISS task, each one resulting from identical processing of a different input set of interactions. Thus, the same approach was used to extract passages for the 3 runs, but the extracted interactions were obtained from different runs of our PPI-IPS task, as described in Section 2.2. The results of each of the runs are presented in Table 1. Some passages were judged as false positives when in fact the passages could be alternative to the passages used in the gold standard for evaluation, as noted by the BioCreAtIVe organizers in the `readme` file of the ISS subtask. The inclusion of such alternative statements will impact our performance positively by reducing the number of false positives.

Table 1. PPI-ISS results. Different sets of interactions were obtained from different runs of our PPI-IPS task. The “Mean” column represents the average performance of all of the BioCreAtIVe PPI-ISS runs.

	Mean	Run 1	Run 2	Run 3
Fraction correct (best) from predicted passages	0.0473	0.0514	0.0483	0.0605
Fraction correct (best) from unique passages	0.0473	0.0496	0.0456	0.0533
Mean reciprocal rank of correct passages	0.5574	0.5731	0.5813	0.5476

We further analyzed 35 out of the 169 true positive passages with respect to their paragraphs of origin. A total of 26 out of the 35 originated from the results section, while 7 passages were from figure captions. This suggests that the intuition behind criteria (1) of passage scoring is reasonable. For criteria (2), the length of the passages was not considered so that it gives higher preferences to long passages over short passages.

Recall that paragraphs stored in the XML format are not necessarily assigned to the actual sections in the original format (due to variations in the section names). The deficiencies of the conversion can affect the scoring of the passages, since scoring is partly based on their origin. To get an approximation of the performance impact of the conversion step, we quantified the converted articles that were incorrectly converted into XML as follows: if (1) there was no text in any of the sections or (2) there were fewer than 5 paragraphs in the references section, the article was flagged as incorrectly converted. Either condition points to a conversion error, since all paragraphs should belong to a section, and articles usually cite more than 5 papers. Of the 358 articles provided as the PPI testing dataset, 48 of the converted articles failed the first condition, and 82 failed the second, indicating a potential “infiltration” of references as regular paragraphs.

Thorough quantification of these problems and their impact in the overall performance of the system is ongoing. Other limitations of our approach reflect the categories identified in [24] as common challenges: (a) discriminating the polarity of passages (b) evaluating the certainty of passages, briefly discussed next.

Discriminating the polarity of passages. Our current approach cannot distinguish if interactions are confirmed or not from the extracted passages. Consider for example the following sentence from PMID 16234233, which should not have been provided as evidence of an interaction:

Passage 1: *“We have not been able to confirm the specificity of the commercially available antibodies against ASIC3 on DRG tissues isolated from ASIC3-inactivated mice.”*

Evaluating the certainty of passages. Some of the passages extracted by our system are mere speculation of hypotheses, and should not have been regarded as correct evidence passages. Consider the following sentences extracted from PMID 16278218:

Passage 2: *“Forced expression of MAPKAP kinase 2 (MK2) appears to lead to phosphorylation of free Heat shock transcription factor 1 (HSF1) on serine 121, and this is associated with HSP90 binding and inhibition of heat shock elements (HSE) binding.”*

Passage 3: *“We have shown that MAPKAP kinase 2 (MK2) directly phosphorylates Heat shock transcription factor 1 (HSF1) and inhibits activity by decreasing its ability to bind the heat shock elements (HSE) found in the promoters of target genes encoding the HSP molecular chaperones and cytokine genes.”*

A human reader can easily distinguish the “we have shown” in Passage 3 as much stronger than the “appears to lead” in Passage 2, but the distinction is not obvious using the scoring criteria of our system.

2.2 Protein Interaction Pairs (PPI-IPS)

The PPI-IPS runs by our group were completed using two natural language processing (NLP) extraction systems, IntEx [25] and Phoenix, that differ in their extraction method but share a number of pre- and post-processing techniques. For both, each paragraph in the source article is broken into individual sentences, which are processed individually. Each sentence is first cleaned by the Jericho HTML Parser [26] that transforms HTML character references into the corresponding ASCII characters. ABNER [21] is then used to identify protein name mentions in the sentence. If at least two protein names and an interaction word from the IEPA corpus [7] are detected, the sentence is parsed by Link Grammar[27], a deep syntactic parser that generates constituent trees and grammatical linkages between words. The differences in the architecture are detailed next, followed by an analysis of our results in this task.

2.2.1 PPI-IPS Architecture

IntEx uses complex combinations of Link Grammar[27] word-to-word linkages to identify subjects -*S*-, objects -*O*-, verbs -*V*-, and modifiers -*M*- in a sentence, and extracts interactions based on patterns of these roles. IntEx has been described in detail in [25]. Phoenix, still under development, is our follow-up system. The main motivating factors for writing a new system were flexibility and extensibility: Phoenix is modular in design, and will be easy to upgrade and fine-tune.

Extracting triplets of interest. An ad-hoc query language was developed to express the rules that detect syntactic roles of words in parsed sentences. The extraction rules use the constituent tree representation provided by Link Grammar to detect subjects, verbs phrases, and objects in each clause of the sentence (rather than the word-to-word linkages used by IntEx). Using the constituent trees facilitates the construction of potentially useful grammatical combinations that result in triplets of the form <subject, verb phrase, object>. These are then filtered to include only protein-protein triplets of interest. As seen in the sample rules in Figure 2, the extraction rules examine the relationships (child, descendent, or sibling) between tree nodes and are used to match patterns of constituents in the tree.

Selecting triplets. In both Phoenix and IntEx, the subject and object are first normalized to their UniProt identifiers using the algorithm described in Section 2.3, attempting to map them first to the most common organisms (humans, yeast, and mouse). If a high-confidence match is not found, then the entire list of UniProt identifiers provided by BioCreAtIVe is used. The triplet-filtering step also uses a list of protein types [5] to strip the type from subjects and objects to prepare protein names for normalization.

Once interactions have been normalized, all the triplets produced by IntEx are used in the final output, whereas Phoenix filters them as follows:

- Remove interactions where both entities are identical
- Keep only one copy of interactions detected multiple times in the same sentence
- Score interactions based on different factors, such as the section where it appears, the number of times the entities and the interaction itself appear, and the confidence level of the normalization step.

The interactions are then sorted by their scores, which are used to decide which interactions to include in Phoenix's output. High precision runs can be created by only outputting interactions with a score greater than a certain threshold.

Evaluation. To aid in our development, we modified our existing prototype web-based evaluation system to support the BioCreAtIVe IPS and ISS submission formats, adding features to aid in rapid evaluation of Phoenix. Like many simple evaluation scripts, the online evaluation system automatically calculates the precision and recall of an uploaded run based on a set of gold standard facts. In addition, it allows a document by document view of each interaction, with the system score for each, plus the source sentence and the extracted protein names before normalization. Throughout development, we could quickly locate incorrectly extracted facts and identify the general source of the error by examining how the protein names were normalized and the grammatical structure of the source sentence. This significantly reduced the time required to assess the effects of changes to the extraction algorithm and was a great aid in determining which areas of Phoenix required improvement.

2.2.2 PPI-IPS Analysis

For the first run, we used Phoenix and tuned the interaction score threshold to try to optimize the *f*-score of the extracted interactions. The second run was also Phoenix, but with a lower threshold to generate more interactions. These interactions were then post-processed to leave only those for which supporting

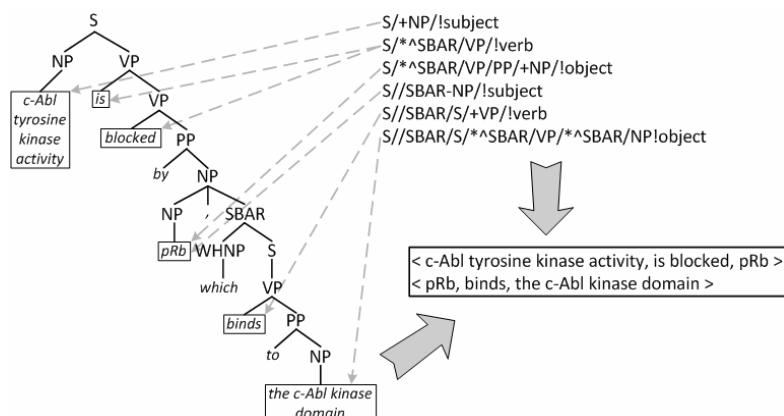


Figure 2. A partial list of extraction rules expressed in the ad-hoc grammar query language developed for our NLP text extraction system (Phoenix), as they apply to a Link Grammar constituent tree.

experimental evidence was found (using the output from the PPI-ISS subtask, described in Section 2.1). Interactions without supporting passages indicating the experimental techniques were pruned. Run 3 was extracted by IntEx without any experimental evidence post-processing. As seen in Table 2, the official BioCreAtIvE evaluation results of our submission, Runs 1 and 3 outperformed Run 2, with Run 3 as the best run overall. Although Run 3 (IntEx) did slightly better, the difference with respect to Run 1 (Phoenix high precision) is not statistically significant, having an effect size of less than 0.02 (negligible).

Table 2. Official scores by run

Protein name normalization was a significant source of error across all three runs. Even with a flawless NLP extraction technique, Table 3 gives the BioCreAtIvE evaluation of the normalization of our predicted interactor proteins. This data shows that even if all pre-normalization extraction modules hypothetically performed flawlessly, our extraction systems' results would still be limited by our ability to map protein name mentions to UniProt[28] identifiers. We further discuss this problem in Section 3.

The blind two-tiered approach used for normalization within this task, where normalization to common organisms is done first, proved problematic. It helped give greater weight to the most common cases, but it introduced errors in others. For example, in one case, a correctly extracted interaction pair was normalized to human proteins instead of yeast, even though his article's title alone, "The Cap-binding protein eIF4E promotes folding of a functional domain of yeast translation initiation factor eIF4G1", shows that IF4E, IF4G1 should be mapped to yeast proteins. Thus, contextual clues need to be examined when selecting the correct organism.

Phoenix relied on ABNER [21] for protein name mentions for sentence classification and triplet filtering. Using the model trained on the BioCreAtIvE corpus, which is what was used, ABNER reports 65.9% recall. Therefore, assuming independence of protein name recognition and ignoring the possibility that a false positive is identified, there is a 56.6% ($100\% - 65.9\% * 65.9\%$) chance that the sentence will be ignored because both protein names in are not recognized. In addition, a single false positive from ABNER could cause multiple false positives in the extracted interaction pairs if the incorrect protein name was present in multiple interaction pairs.

We traced most of the remaining errors to Link Grammar and the rules used to extract interaction pairs from its constituent tree output. At the time of submission, Link Grammar split multiword protein names when building a constituent tree. This made normalization of the interaction pairs much more difficult, but has since been corrected. Moreover, Link Grammar produces many possible linkages and constituent trees for each sentence, but the first linkage and constituent tree returned by Link Grammar was always used for the extraction. Upon manual examination, it was found that the first linkage and tree returned were not always the best representation of the sentence structure. In addition, much of the information to be gained by using a deep parse instead of a shallow POS tagging was not exploited. In Phoenix, subjects, verb phrases, and objects were grouped into sets and combined based on the clause of the sentence that contained them, rather than the tree structure. The rules themselves covered only the most general sentence constructs, which led Phoenix to overlook protein interactions expressed in less common grammatical forms. These issues are presently being addressed in the refinement of the Phoenix extraction system.

2.3 Gene Normalization

The gene normalization system we implemented was a lightweight implementation which mixed well-known systems with the implementation of new, relatively nonstandard, ideas. Overall, the system relied heavily on orthographic and syntactic information rather than semantic knowledge, including biological domain knowledge. Its architecture and analysis of results follow.

2.3.1 Architecture

The Gene Normalization Task receives as input an abstract to process and produces a list of normalized gene mentions from the text. The system completes 4 distinct execution phases: extraction, filtering, normalization

	Run 1 Phoenix	Run 2 Phoenix	Run 3 IntEx
Mean Precision	0.0456	0.020172	0.056049
Mean Recall	0.124279	0.099706	0.136227
Mean F-score	0.055964	0.029517	0.068575
Overall Precision	0.036957	0.020233	0.052997
Overall Recall	0.080189	0.069575	0.071934
Overall F-score	0.050595	0.03135	0.061031

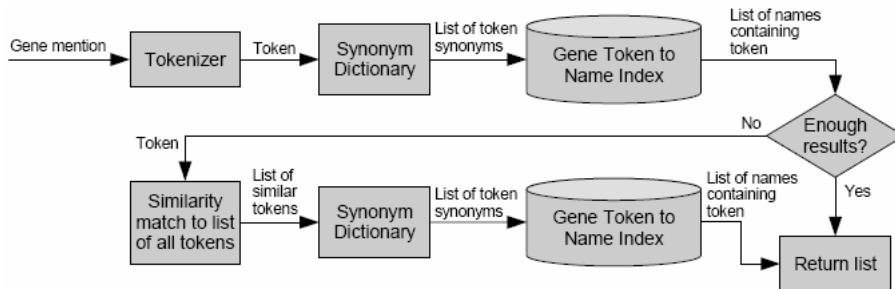


Figure 4. Gene normalization system. Gene mentions are compared first as a complete instance, and then at the token (word) level if not enough matches are found.

and disambiguation, with most of the complexity residing in the normalization phase. There, each gene mention is tokenized and compared against the standard gene names and a similarity score is computed for each. A list of the most similar standard gene names is then returned. We describe details of each phase next.

Extraction. We intended the system to primarily test gene normalization ideas and therefore employed the same ABNER [21] system for tagging gene mentions in each abstract, and as for the other tasks, used the model trained on the BioCreAtIvE 1a task. After gene mentions are tagged and extracted, acronyms are resolved using the Stanford Biomedical Abbreviation database, described in [29], and their provided Java code. The list of gene mentions found is the only data passed from the abstract to the next phase.

Filtering. In the filtering phase, mentions of generic words (such as “gene” and “protein”) are dropped. Specifically, gene mentions which consist entirely of generic words are removed; all other mentions are retained. The list of generic words contains about 100 entries of the following types:

- Organism names such as “yeast”, “human”, and “E. coli”
- General protein types and descriptors like “enzyme”, “amyloid”, and “protein”
- Other terms related to molecular biology, but not gene names, such as “DNA” or “alpha”

Normalization. Each gene mention which passes filtering is capitalized and separated into tokens. The system then compares the mention with each of the standard gene names and computes a similarity score for each comparison. This score is based on the Dice coefficient [30], and therefore reflects the number of tokens contained in both the gene mention and the standard gene name, scaled to reflect the lengths of both, and gives twice the weight to agreements. A perfect match has a similarity score of 1.0 while the similarity score for an attempted match with no tokens in common is 0. The equation for the standard Dice coefficient is

$$dice(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}.$$

The standard calculation was modified in the following ways:

- Each token was given a weight based on the frequency with which it appears in the list of gene names. Tokens appearing more frequently have a lower weight than tokens appearing less frequently, according to the following function $w(x) = 1 - \left(\frac{f(x)}{a \times m}\right)^{\frac{1}{a}}$, where $f(x)$ is the frequency of the token in the list of gene names, m is the maximum frequency of any token and a is an empirically-determined tuning parameter greater than 1. Note that for any token x , $0 \leq w(x) \leq 1$. This weighting scheme was designed to decrease more slowly than simply using the inverse of the token frequency.
- The Dice coefficient is further modified to give tokens from the gene mention a higher weight than tokens from the gene name. This reflects the fact that the gene mentions have, on average, fewer tokens than the standard gene names.

$$\text{These modifications result in the equation } dice_w(X, Y) = \frac{2 \times \sum_{z \in X \cap Y} w(z)}{a \times \sum_{x \in X} w(x) + (1-a) \times \sum_{y \in Y} w(y)}.$$

Tokens are initially considered a match if they contain exactly the same series of characters or represent synonymous ordinal values, such as Arabic and Roman numerals and the letters of the Greek alphabet.

To boost precision, thresholding is applied so that matches with a low score are dropped from further consideration. A list of candidate gene names taken from the top matches is then associated with each gene mention as it moves into the disambiguation phase.

Disambiguation. Since the normalization phase returns a set of candidate gene names from the standard list, it is necessary to determine which of the candidates is the most likely to be correct. Disambiguation proceeds in a short series of automated steps based on simple rules as follows:

1. Gene mentions where the similarity margin – the difference between the similarity of the best match and the similarity of the second best match – is above a threshold are considered unambiguous. For these, the genes to which the best-matching gene name refers are added to the final output. The margin threshold used is preset and was determined empirically using the training set.
2. Gene mentions which remain after step 1 are reviewed to determine if their list of potential matches contains a name which refers to a gene already accepted as unambiguous. The intuition is that the abstract is most likely referring to the same gene by different names. The gene mention is removed.
3. Finally, for any remaining gene mentions, the best-matching gene name is accepted and the gene to which it refers is added to the final output.

2.3.2 Gene Normalization Analysis

The system achieved a recall of 0.713 and a precision of 0.520 on the test set, for an f-measure of 0.602. We believe that these results demonstrate that metric-based methods are insufficient, even when coupled with orthographic similarity between two tokens. Table 3 shows the evaluation of several variants of the system, showing the respective contribution of the various phases.

Table 3. Adjusted performance measures on GN system variations.

Variation	Precision	Recall	F-Measure
As evaluated for the competition	0.462	0.667	0.546
Without filtering phase	0.440	0.670	0.531
Standard Dice coefficient instead of weighted	0.461	0.669	0.546
No threshold-based removal of low similarity matches	0.339	0.713	0.460
Return best match instead of using disambiguation rules	0.439	0.692	0.537

Using acronym resolution to substitute the original text of the gene mention introduces a problem when the standard gene names also contain abbreviations.

The simple disambiguation rules used to eliminate generic mentions perform reasonably well in practice, and their failures are generally due to failures in the normalization to correctly identify semantic equivalence. However, the current method of relying on a small dictionary is brittle and ought to be based on a wide sampling of molecular biology terms. A more flexible method may be to perform filtering after the normalization step by noting that generic mentions are going to match a wide variety of standard gene names at a low level of similarity, but match none of them well.

3 Discussion

Three important developments from our participation include the development of the overall architecture that allows a more flexible incorporation of the different components using a standard input/output XML format, the development of a new extraction system flexible enough to sustain generic extractions of relationships in biomedical text, and the development of a flexible evaluation platform. Given the reliance of the overall knowledge extraction and integration approach on solid gene mention and gene normalization modules, these two subsystems will occupy a good part of our efforts.

For the extraction of related statements (evidence of interaction being one of them), we will expand on the issues of polarity and certainty of passages, as they are critical to the problem of finding passages with experimental evidences.

As for the extraction system, future development will initially focus on improving the manner in which the extraction rules are used to identify potential interactions. The algorithm that combines the subjects, verbs, and objects will be modified to utilize the relationships between these syntactic roles by analyzing their common ancestors in the constituent tree. Furthermore, we have learned that organism identification is a nontrivial component of successful protein name normalization. Before normalizing, we will search for context clues regarding the organisms and provide this information to the normalization process.

References

- [1] "Pathguide: The Pathway Resource List."
- [2] E. S. Soteriades and M. E. Falagas, "Comparison of amount of biomedical research originating from the European Union and the United States," *BMJ: British Medical Journal*, vol. 331 pp. 192-194, 2005.
- [3] Z. Lu, K. B. Cohen, and L. Hunter, "Finding GeneRIFs via Gene ONtology Annotations," presented at Pacific Symposium on Biocomputing, Maui, Hawaii, USA, 2006.
- [4] L. Tanabe and W. J. Wilbur, "Tagging gene and protein names in biomedical text," *Bioinformatics*, vol. 18, pp. 1124-1132, 2002.
- [5] G. Leroy, Chen, H. , et al., "Genescene: biomedical text and data mining," presented at The third ACM/IEEE-CS joint conference on Digital libraries, 2003.
- [6] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi, "Automated extraction of information on protein-protein interactions from the biological literature," *Bioinformatics*, vol. 17, pp. 155 - 161, 2001.
- [7] J. Ding, Berleant, D., Xu, J., Fulmer, A., "Extracting biochemical interactions from MEDLINE using a link grammar parser," *IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03)*, pp. 467, 2003.
- [8] S. T. Ahmed, D. Chidambaram, H. Davulcu, and C. Baral, "IntEx: A Syntactic Role Driven Protein-Protein Interaction Extractor for Bio-Medical Text. , " in *BioLINK SIG: Linking Literature, Information and Knowledge for Biology, a Joint Meeting of The ISMB BioLINK Special Interest Group on Text Data Mining and The ACL Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics (Biolink'2005)*. Detroit, Michigan, 2005.
- [9] T. M. Phuong, Lee, D., Lee, K. H., "Learning Rules to Extract Protein Interactions from Biomedical Text," *PAKDD 2003*, pp. 148-158, 2003.
- [10] A. Koike, Y. Niwa, and T. Takagi, "Automatic extraction of gene/protein biological functions from biomedical text," *Bioinformatics*, vol. 21, pp. 1227-1236, 2005.
- [11] H.-W. Chun, Y. Tsuruoka, J.-D. Kim, R. Shiba, N. Nagata, T. Hishiki, and J. i. Tsujii, "Extraction of Gene-Disease Relations from Medline Using Domain Dictionaries and Machine Learning," presented at Pacific Symposium on Biocomputing, 2006.
- [12] C. Perez-Iratxeta, P. Bork, and M. Andrade, "Association of genes to genetically inherited diseases using data mining," *Nature Genetics*, vol. 31, pp. 316-319, 2002.
- [13] D. Hristovski, B. Peterlin, J. Mitchell, and S. Humphrey, "Improving literature based discovery support by genetic knowledge integration," *Stud Health Technol Inform 2003*, vol. 95, pp. 68-73, 2003.
- [14] T. C. Rindflesch, L. Tanabe, J. N. Weinstein, and L. Hunter, "EDGAR: extraction of drugs, genes and relations from the biomedical literature," *Pac Symp Biocomput*, pp. 517 - 528, 2000.
- [15] A. S. Yeh, L. Hirschman, and A. A. Morgan, "Evaluation of text data mining for database curation: lessons learned from the KDD Challenge Cup," *Bioinformatics*, vol. 19, pp. i331-339, 2003.
- [16] H. Hermjakob, L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman, and R. Apweiler, "IntAct: an open source molecular interaction database," *Nucl. Acids Res.*, vol. 32, pp. D452-455, 2004.
- [17] A. Zanzoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni, "MINT: a Molecular INTeraction database," *FEBS Letters*, vol. 513, pp. 135-140, 2002.
- [18] I. Donaldson, J. Martin, B. de Bruijn, C. Wolting, V. Lay, B. Tuekam, S. Zhang, B. Baskin, G. Bader, K. Michalickova, T. Pawson, and C. Hogue, "PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine," *BMC Bioinformatics*, vol. 4, pp. 11, 2003.
- [19] "Lucene."
- [20] A. Schwartz and M. Hearst, "A simple algorithm for identifying abbreviation definitions in biomedical texts," *In Proceedings of the Pacific Symposium on Biocomputing (PSB 2003)*, vol. 8, pp. 451-462, 2003.
- [21] B. Settles, "ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text," *Bioinformatics*, vol. 21, pp. 3191-3192, 2005.
- [22] "Molecular Interaction (MI) ontology."
- [23] M. F. Porter, "An algorithm for suffix stripping..," *Pro-gam*, vol. 14, pp. 313--316, 1980.
- [24] W. J. Wilbur, A. Rzhetsky, and H. Shatkay, "New directions in biomedical text annotation: definitions, guidelines and corpus construction," *BMC Bioinformatics*, vol. 7, pp. 356, 2006.
- [25] S. Ahmed, D. Chidambaram, H. Davulcu, and C. Baral, "Intex: A syntactic role driven protein-protein interaction extractor for bio-medical text," *Proceedings ISMB/ACL Biolink*, pp. 54-61, 2005.
- [26] "Jericho HTML Parser."
- [27] D. Sleator and D. Temperley, "Parsing English with a Link Grammar," *Third International Workshop on Parsing Technologies*, 1993.
- [28] "UniProt."
- [29] J. D. Wren, J. T. Chang, J. Pustejovsky, E. Adar, H. R. Garner, and R. B. Altman, "Biomedical term mapping databases," *Nucl. Acids Res.*, vol. 33, pp. D289-293, 2005.
- [30] L. Egghe and C. Michel, "Strong similarity measures for ordered sets of documents in information retrieval," *Information Processing and Management*, vol. 38, pp. 823-848, 2002.



Mining Physical Protein-Protein Interactions by Exploiting Abundant Features

Minlie Huang^{1,\$,*}
aihuang@tsinghua.edu.cn

Shilin Ding^{1,\$}
dingsl@gmail.com

Hongning Wang^{1,\$}
whn03@mails.tsinghua.edu.cn

Xiaoyan Zhu¹
zxy-dcs@tsinghua.edu.cn

¹ State Key Laboratory of Intelligent Technology and Systems (LITS), Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China

* Corresponding author: Email aihuang@tsinghua.edu.cn; Fax +86-10-62782266; Tel +86-10-62796260

Abstract

In this paper, we present approaches of mining physical protein-protein interactions by exploiting abundant features during our participation in the PPI task of BioCreAtIVe Challenge 2006. In the first task of classifying whether an article contains at least one physical protein-protein interaction, a feature-based and kernel-based SVM and probabilistic model have been studied, where abundant features, including strings, unigrams, semantic features from external resources, are exploited. In the second task of extracting interacting protein pairs, we proposed a *profile*-based method which adopts position feature, template feature, and term feature. The method extracts interactions at the document level, which will be less influenced by errors caused by named entity recognition. In the third task, models in the previous tasks are integrated together to extract and rank summary sentences. Compared with the mean performance averaged across all teams, our method has shown to be very competitive.

Keywords: protein-protein interaction, relation extraction, named entity recognition, SVM, kernel

1 Introduction

It is challenging to mine protein-protein interactions from bioscience literature. From a general perspective, there are three sub-tasks to mine *biologically meaningful knowledge*: first, classify whether or not a document contain interactions; second, extract protein-protein interactions (or interacting protein pairs) from relevant documents; finally, extract detailed information about interactions, such as experimental detection methods of interactions, and summary sentences describing them. Characterizing protein interaction partners is crucial to understanding not only the functional role of individual proteins but also the organization of entire biological processes. Although there are databases storing molecular interactions [1] [2], most of them are still hidden in literature. Motivating the implementation of information extraction techniques, a number of approaches have been published [3] [4] [5].

We participated in three sub-tasks of the Protein-Protein Interaction task in BioCreAtIVe Challenge 2006 [6]. The first one is to classify whether or not a given article contains at least one *physical* protein-protein interaction. This has often been neglected by previously published protein-interaction extraction systems. Given a number of articles, the participants are required to return two ranked lists of articles, where one list contains physical interactions, the other not. The second sub-task to extract a ranked list of protein pairs from full text articles, where each protein pair is claimed to interact physically by the article. The third one is to find a ranked list of maximum five passages describing interactions between proteins. These tasks are very difficult because 1) we have to map protein entities into Uniprot IDs (or SwissProt IDs), 2) the articles are full text articles in html format, and 3) annotation information is not given at the sentence level but only at the document level.

^{\$} We want readers to know that these authors have equal contributions to this work.

In this paper, we describe approaches to mine physical protein-protein interactions by exploiting abundant features. In the first task of classifying whether a document contains physical interactions, p -spectrum kernel based *SVM*, feature-based *SVM*, and multinomial probabilistic model are proposed, and several types features including string, unigram, and bigram, are exploited. In the second sub-task, a template-based and *profile-based* method is adopted to extract protein pairs from full text articles. In profile-based method, we extract interactions at the document level, and abundant features such as position feature and template features, are employed. The models in these two steps are integrated together to extract summary sentences for each interactions. In comparison to the mean performance released by official organizers, our method shows promising results.

2 Methods and Results

2.1 Sub-task I: Classifying and Ranking Articles

Given a set of MEDLINE abstracts, the task firstly requires classifying whether or not an article contains at least one *physical* protein-protein interaction, and secondly needs to rank relevant and irrelevant documents separately, according to the confidence of the prediction. A *relevant* document here means there is at least one PPI in it. In this task, the abstract and title of an article are concatenated together, and other information in the training texts is omitted.

Three classifiers are studied here: p -spectrum kernel based *SVM*, feature-based *SVM*, and multinomial probabilistic model. In the first model, an article is treated as a string, and all common sub-strings of length p are computed for every two articles, as defined by the following formulas:

$$K_p(x, y) = \langle \phi^p(x), \phi^p(y) \rangle = \sum_{u \in \Sigma^p} \phi_u^p(x) * \phi_u^p(y) \quad (1)$$

$$\phi_u^p(x) = \left| \{(v_1, v_2) \mid x = v_1 u v_2, u \in \Sigma^p\} \right| \quad (2)$$

where x and y are two strings (or documents) defined on alphabet Σ , and Σ^p indicates all possible sub-strings of length p . The alphabet used in our experiment is the 26 English characters plus one white-space. The computation of such a kernel is very simple, however, a naïve implementation will cost a large amount of computational complexity ($O(p * |x| * |y|)$). In our experiments, we implement the algorithm using *trie-tree* structure which reduces the complexity to $O(p * (|x| + |y|))$. The length of p is set to 5, 6, or 7.

The second model we studied is feature-based *SVM* (with linear kernel), where abundant features have been exploited. Each article is represented a feature vector. There are three types of features:

- (1) unigram features selected by chi-square statistics,
- (2) features from Molecular Interaction Ontology (MIO) [7],
- (3) and features from Enzyme Nomenclature (EN) [8].

Note that the task requires extracting articles containing *physical* interactions, but not *genetic* interactions or anything else. Interaction type is defined explicitly by MIO, where there are *colocalization*, *genetic*, and *physical* interactions. This implies there are useful features to discriminate *physical* interactions from non-physical ones. Unigrams that are statistically significant, are extracted from each node (including name, definition, description) in the branch of *interaction type*. Moreover, we select features from the branch of *interaction detection method*, where there are terms strongly indicating physical interactions, for instance, *two-hybrid*. We select features from EN because there is a branch of *enzymatic reaction* inherited from *physical interaction*. It is a strong sign for physical interaction with the presence of an enzyme name or enzyme suffix. Suffix words such as *synthetase* and *translase* are extracted from the dictionary of EN. We call features from MIO and EN *semantic features*, because they reflect a semantic link to physical interaction.

The above two models are discriminative models which learn a decision hyper-plane to classify samples. However, we need not only classify, but also rank samples. The distance of a sample from the decision hyper-plane can be used for ranking. In the framework of *SVM*, we know the decision function is as below:

$$f(x) = \text{sgn}(b + \sum_{i \in SV} y_i \alpha_i K(x_i, x)). \quad (3)$$

And the distance of a sample x from the decision hyper-plane is a constant ratio of the term:

$$R(x) = b + \sum_{i \in SV} y_i \alpha_i K(x_i, x) \quad (4)$$

For relevant articles, $R(x)$ can be used to rank documents; for irrelevant articles, $-R(x)$ is used instead. In our

submitted results, a slightly modified decision function is used:

$$c_x = \begin{cases} c_{+1}, & R(x) > \Delta \\ c_{-1}, & \text{otherwise} \end{cases} \quad (5)$$

where Δ is a threshold specified by cross-validation experiments.

The third model is a probabilistic model, which estimates a probability distribution on a set of random variables. The basic idea is that articles can be ranked by the likelihood of being a positive sample. Here an article is viewed as a bag of features $\{w_1, w_2, \dots, w_n\}$, and each feature w_i appear x_i times. Suppose there be a multinomial distribution to generate an article from the features, and then we have the following score to rank documents:

$$\begin{aligned} R(d) &= \log \Pr(c_{+1} | d) - \log \Pr(c_{-1} | d) \\ &= \log \Pr(d | c_{+1}) + \log \Pr(c_{+1}) - \log \Pr(d | c_{-1}) - \log \Pr(c_{-1}) \\ &= \sum_{i=1}^n x_i \log \Pr(w_i | c_{+1}) - \sum_{i=1}^n x_i \log \Pr(w_i | c_{-1}) + \log \Pr(c_{+1}) - \log \Pr(c_{-1}) \end{aligned} \quad (6)$$

where d is a document, c_{+1} indicates relevant documents while c_{-1} irrelevant. Each probability is estimated by

$$\Pr(w_i | c_{+1}) = \frac{N(w_i, c_{+1}) + 1}{V + \sum_{w_i} N(w_i, c_{+1})} = \frac{1 + \sum_{d_j \in POS} tf(w_i, d_j)}{V + \sum_{w_i} \sum_{d_j \in POS} tf(w_i, d_j)}. \quad (7)$$

Here V is the total number of unique features, $N(w_i, c_{+1})$ is the total times of feature w_i appearing in relevant documents, POS is the set of all relevant documents, and $tf(w_i, d_j)$ is the term frequency of w_i in document d_j . $\Pr(w_i | c_{-1})$ can be calculated similarly by substituting c_{-1} for c_{+1} in the formula. The decision function for classification is defined by

$$c_d = \begin{cases} c_{+1}, & R(d) > \Delta_{th} \\ c_{-1}, & \text{otherwise} \end{cases}. \quad (8)$$

where Δ_{th} is an experimentally determined threshold, and c_d is the class label of document d . The decision function is firstly used to classify relevant documents from irrelevant ones and then $R(d)$ is used for ranking.

2.1.1 Experiment and Discussion

There are totally 3536 articles relevant to physical interactions and 1959 irrelevant. Although there are additional articles provided (noisy, some are describing genetic interactions), our experiments show worse performance if noisy data are used. Hence we did not use this part of data. To determine the thresholds used in Formula (5) and (8), these articles are divided into four parts, and 4-fold cross validation is performed. The threshold is set when the best F_1 score is obtained on the leave-out part of articles.

Thresholds for feature selections are:

- (1) auto-mined features: the total frequency in training data >50, chi square value >3.84;
- (2) features from MIO: the total frequency >20, chi square value >3.84;
- (3) features from EN: the total frequency >20, chi square value >3.84.

The results are shown in Tab. 1. The best performance of ours is obtained by the p -spectrum kernel based SVM. This is very surprising because only very low-level features are used in this model, and we did not consider any semantic level features. In comparison, the other two models employed high-level features, some of which are incorporated from semantic aspects, for example, those selected from MIO and EN. Another problem to be analyzed is that the performance evaluated on the official test dataset is much worse than that on previous released data. For example, we achieved a precision of 93% and a recall of 94% with p -spectrum kernel ($p=7$) during a four-fold cross validation.

To analyze these problems, 750 articles (375 positive) are randomly taken out of the training corpus as a leave-out test dataset (LOD for short). The top 50 features whose significance is measured by chi-square statistics, is selected from the remaining training dataset (RTD for short). Based on the 50 features, three probability distributions are estimated on LOD, RTD and official test dataset (OTD for short), by using formula (7). Then we compute the average Kullback Leibler divergence between two distributions to measure how different two distributions are, as follows:

$$AKL(q, p) = \frac{1}{2} \sum_x (q(x) \log \frac{q(x)}{p(x)} + p(x) \log \frac{p(x)}{q(x)}). \quad (9)$$

These results are presented in Tab. 2. For $Pr(x|c_{+I})$, there is no significant difference between term distributions estimated on RTS, LOT, or OTS. In other words, the three different data sets have almost an identical term distribution. However, the case is significantly different for $Pr(x|c_{-I})$ where distributions are illustrated by Fig. 1. There is a large divergence (0.99) between the distribution estimated on the official test set and that on training data set. We conjecture that there is a different term distribution on the official test set, and this may be the reason why the model degraded markedly on the official data set. When string is selected as feature, as p -spectrum kernel does, the divergence is much less (0.992 vs. 0.188). This might explain why string feature even excelled term feature in these runs.

Table 1: Average results over 51 runs from 19 teams. AUC is the area under ROC curve.

Score	Precision	Recall	F-score	AUC	Accuracy
Mean	0.6642	0.7636	0.6868	0.7351	0.6705
Standard Deviation	0.0810	0.1926	0.1035	0.0741	0.0644
Median	0.6772	0.8507	0.7224	0.7515	0.6680
p -spectrum kernel ($p=7$)	0.7352	0.8293	0.7794	0.8375	0.7653
Feature-based SVM	0.7333	0.7920	0.7615	0.8127	0.7520
Probabilistic Model	0.6855	0.8080	0.7417	0.8034	0.7187

Table 2: Average KL divergence between distributions on different data sets. Dis. = distribution

Compared Distributions	Term Feature		String Feature	
	$Pr(x c_{+I})$	$Pr(x c_{-I})$	$Pr(x c_{+I})$	$Pr(x c_{-I})$
Dis. on RTD vs. Dis. on LOD	0.0216	0.0703	0.0029	0.0163
Dis. on RTD vs. Dis. on OTD	0.0369	0.9926	0.0357	0.1887

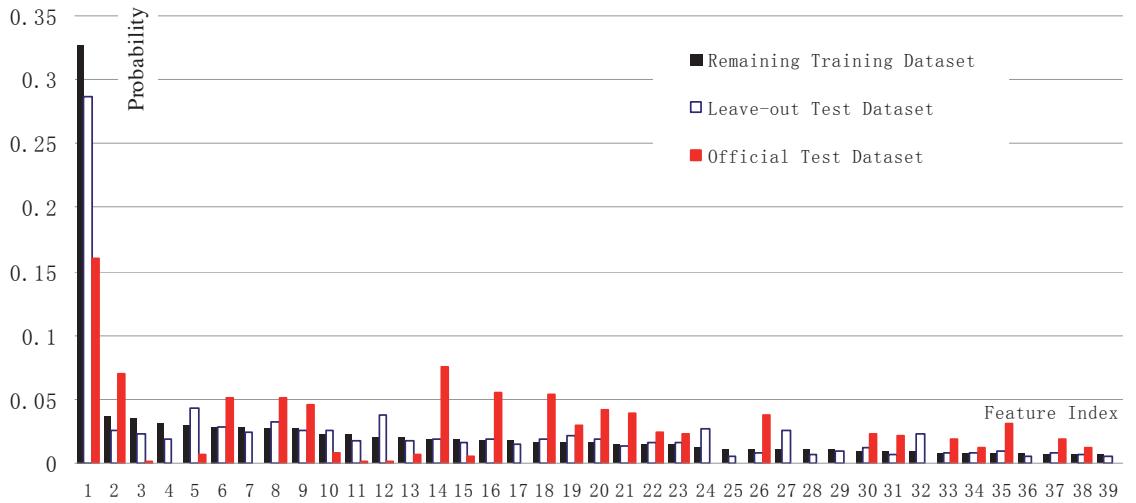


Figure 1: $Pr(x|c_{-I})$ on different datasets (only 40 features are listed here).

2.2 Sub-task II: Extracting Protein Pairs from Full Text Articles

The goal of this task is to identify physically interacting protein pairs from full text articles. There are two major challenges here: 1) recognizing protein named entities and mapping each entity to a unique entry in the UniProt database; 2) identifying protein pairs which have been experimentally confirmed to have physical

interactions.

The framework of our system is illustrated in Fig. 2. There are three modules: preprocessing module, named entity recognition (NER) module, and protein-protein interaction identification module.

The Preprocessing module is mainly concerned with extracting useful text contents from full text articles in HTML format. First, a HTML Parser [9] is applied to extract useful contents. Then the text is organized in XML format, where titles, abstracts, subtitles, captions, paragraphs are well indicated by XML tags. Encoding problems, such as images with “ALT” tag and unicode characters, for instance, mathematical symbols and Greek characters, are solved to make the plain text clean and well-organized. To remove irrelevant contents more efficiently, sub-modules are constructed according to different HTML styles designed by different journals.

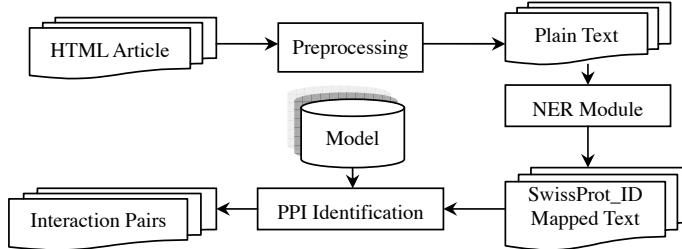


Figure 2: the framework for extracting protein pairs. NER=named entity recognition.

Different from traditional NER tasks, this task requires the submitted protein pairs should be mapped into unique UniProt IDs, instead of presenting the original names in the text. We not only need to recognize named entities but also map them to unique identifiers. As shown in Fig. 3, there are main four processes in this module: database curation, organism detection, dictionary-based matching, and mapped name disambiguation. During database curation, the two steps below are done to improve the quality of terms in SwissProt database:

- Curate entry terms in UniProt records. The gene names/synonyms, gene product names/synonyms of the same entry are included. Addition of gene names may cause ambiguity since a gene may encode several proteins.
- Normalize terms based on rules. Terms except abbreviations are converted to lowercases. Prefixes and suffixes which are not critical for entity identification are removed. For example, prefix *c*, *n* and *a* of PKC (*Protein Kinase C*), which mean *conventional*, *novel* and *atypical* respectively, are removed. And terms including digits or Roman/Greek numbers are transformed into a unified format: Alphabet + white space + digits. This rule implies such normalization: IL-2, IL2, IL 2 → IL 2; CNTFR alpha, CNTFR A, CNTFR I → CNTFR 1.

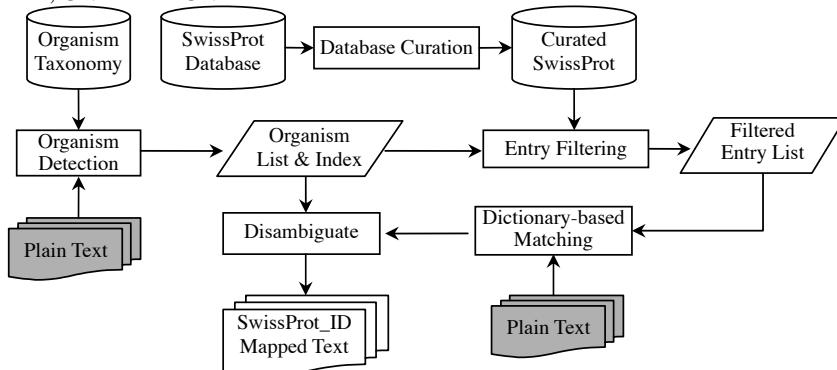


Figure 3: the flowchart of NER module.

After curation, there are totally 230,000 protein IDs, and more than 1 million terms. Obviously, it is not feasible if all the terms are used during dictionary-based matching. Furthermore, the same terms, particularly abbreviations, may correspond to many protein IDs. This is common in cases that many gene products from different organisms are the same. To improve computation efficiency, we first detect the organism information in an article, and then use the information to rule out irrelevant entries and further to remove

ambiguities when different terms are mapped to the same protein identifier. Our assumption here is that physical interactions described in one paper should be within only a few organisms. The organism database used here is NCBI taxonomy [10]. A dictionary-based matching is used to detect organisms, and five most frequent organisms are left. Each sentence is linked with several detected organisms. To disambiguate mapping from recognized names to IDs, the principle of nearest neighbor is used.

In PPI identification module, two models are studied. One is derived from ONBIRES [11] [12], and the other is based on *profile* features. As presented in the original paper, the first model learns lexicon-syntactic templates describing interactions in a semi-supervised manner, and then uses template-matching to extract interactions, where the matching score must exceed a pre-specified threshold. The matching score is also used for ranking protein pairs. In our submitted runs, two different thresholds are attempted. In this model, interactions are extracted at the sentence level. Obviously, this kind of approaches are sensitive to the performance of *NER*, however, as mentioned before, the performance of *NER* is still far from satisfactory in this task.

The second model is a *profile*-based method. The basic idea is to extract interactions by using features derived from the whole document. In other words, we will extract protein pairs at the document level because there are many errors caused by *NER* at the sentence level. And by incorporating document-level information, we can extract interaction pairs more robustly. Every candidate protein pair occurred within a sentence is viewed as a sample. For each pair, *profile-features* are extracted from the whole document. There are several types of features:

- 168 features selected by chi-square statistics
- 91 template features extracted from ONBIRES. Such features have a form as “#protein#*bind to*#protein#”, where #protein# indicates a protein entity, * means any word can be skipped. Template features are matched against sentences.
- 2 position features. One is whether two proteins co-occur within a title; the other is whether co-occurring within an abstract.

These features form a 261-dimensional feature vector, where each dimension is 1 or 0 indicating the presence or absence of a feature. Through such a representation with abundant features, information from the whole document has been incorporated.

2.2.1 Experiment and Discussion

In the first model for interaction identification, we simply adopt templates and algorithms from ONBIRES. In the profile-based model, we construct 2103 feature vectors from provided 740 articles. The official evaluation corpus consists of 358 articles.

Tab. 3 shows the overall performance for both average results over all runs and our submitted results. The *profile*-based model achieves the best results among all the three submitted runs. This model contributes a much better precision than others. Tab. 4 shows the average performance across all articles. Again the *profile*-based method excels others significantly. It is worth noting that our results are much better than the mean performance across all runs from all teams.

We also compared the performance of named entity recognition. These results are shown in Tab. 5. Note that our *NER* performance is much better than the mean performance by database curation, terms normalization, and organism-based disambiguation.

Table 3: Overall performance averaged over 45 runs from 16 teams vs. our overall results

Score	Proteins normalized to UniProt entries			Proteins normalized to only SwissProt entries		
	Precision	Recall	F-score	Precision	Recall	F-score
Mean	0.0938	0.1064	0.0781	0.1015	0.1150	0.0848
Std. Dev	0.0881	0.0704	0.0505	0.0937	0.0755	0.0549
Median	0.0609	0.1097	0.0705	0.0649	0.1179	0.0769
ONBIRES (th=0.0)	0.1191	0.1779	0.1427	0.1333	0.1934	0.1578
ONBIRES (th=80.0)	0.2047	0.1159	0.1480	0.2215	0.1215	0.1569
Profile-feature based	0.2578	0.1097	0.1539	0.2950	0.1179	0.1685

Table 4: Mean performance across articles vs. our mean results

Score	Proteins normalized to UniProt entries			Proteins normalized to only SwissProt entries		
	Precision	Recall	F-score	Precision	Recall	F-score
Mean	0.1062	0.1858	0.1035	0.1160	0.2000	0.1127
Std. Dev	0.0945	0.1001	0.0761	0.1035	0.1062	0.0836
Median	0.0755	0.1961	0.0788	0.0808	0.2156	0.0842
ONBIRES (th=0.0)	0.1373	0.2905	0.1578	0.1566	0.3189	0.1784
ONBIRES (th=80.0)	0.2177	0.2651	0.2038	0.2434	0.2828	0.2247
Profile-feature based	0.3096	0.2935	0.2623	0.3695	0.3268	0.3042

Table 5: Comparative overall results for NER (normalized to SwissProt entries).

Score	Precision	Recall	F-score
Mean	0.1495	0.2828	0.1707
Std. Dev	0.0963	0.1294	0.0764
Median	0.1337	0.2723	0.1683
ONBIRES (th=0.0)	0.2118	0.3816	0.2725
ONBIRES (th=80.0)	0.2618	0.2645	0.2631
Profile-feature based	0.3483	0.2410	0.2849

2.3 Sub-task III: Extracting Summary Sentences for Interactions

This task is to extract summary sentences for each mined interaction. This kind of knowledge is really meaningful to biologists because it will be greatly helpful to understand the underlying biological functions and processes from the summary of a large amount of literature. This task is extremely difficult in that nothing except full text articles is presented to participants. Thus to complete the task, we have firstly to detect interacting protein pairs, and then to extract sentences that could be the summary of an interaction. The system framework is presented in Fig. 4.

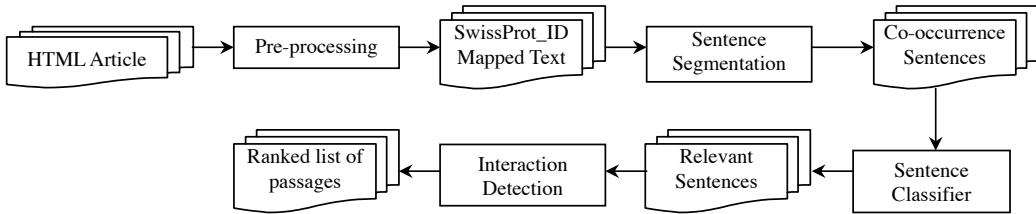


Figure 4: The Framework of Extracting Summary Sentences for Interactions

First, the HTML articles are pre-processed and transformed to plain texts with the protein names being mapped to SwissProt IDs. Then sentences with no less than 2 protein entities, which are termed *co-occurrence sentences*, are selected as candidates for further classification. The techniques used in the pre-processing have been mentioned in Section 2.2.

Second, a SVM classifier based on p -spectrum kernel, which have been trained for the first sub-task as presented in Section 2.1, is used to identify sentences containing protein-protein interactions. The sentences whose output scores from the classifier are below 0 (a tunable threshold) are eliminated. And then the output score of the classifier, S_i , is taken into account ranking summary sentences.

Third, these scored sentences are fed into an interaction detection model to extract the exact protein pairs that have interactions. Two different models are used here: template based model and *profile*-based SVM model, which are both reproduced from the second sub-task as shown in Section 2.2. In the template-based model, we use two sets of templates. The first is automatically selected by the semi-supervised template learning algorithm, and the second is a manually curated version of the first set. Based on the two sets of templates, two runs have been submitted, respectively. The output score of the interaction detection model, S_2 , is considered as another factor for ranking sentences. For the purpose of generating the ranked list of summary passages, the two scores are multiplied in a straightforward manner:

$$S = S_1 * S_2. \quad (10)$$

Finally sentences or passages are ranked by using S . In our method, we do not distinguish between sentence and passage.

The second step is to extract candidate sentences which may contain physical interactions, while the third step is to discover which two proteins are connected by an interaction. The two-step processing will make it portable to extract multi-document summary.

Finally, these sentences/passages are mapped back to the original HTML texts by a novel method based on edit distance. The difficulty here is that there is no tracked information where modifications occur during converting from HTML format texts to plain texts. Here, edit distance between two sentences is used to find original HTML sentences. The rationale behind this method is that the commonly used words are the same in the two sets of sentences (original HTML sentences and processed plain sentences). Details are omitted here.

2.3.1 Experiment and Discussion

Results are shown in Table 6. The first row is the average results over 26 submitted runs. The second column to eighth column are the (average) number of predicted passages, (average) number of correctly predicted passages, (average) number of unique passages, (average) number of correct unique passages, percentage of correct prediction, percentage of correct unique prediction, and mean reciprocal rank (MRR) of correct passages.

Our results are slightly better than the average performance, but much worse in terms of MRR. There may be two major issues in our problem. First, the sentence classifier is training on abstracts plus titles from articles, however, sentences are much shorter than training articles during classification. This is a major gap between learning and classification, which may degrade the performance remarkably. Second, an extremely straightforward schema as defined in Formula (10) is used to take into account factors in the two steps. Unfortunately, the two scores, S_1 and S_2 , may be significantly different from each other. A better solution should normalize them to a comparable range.

Table 6: Averaged results over 26 runs vs. our results

	# Pred(A)	# TP (A)	# Pred(U)	# TP (U)	% (A)	% (U)	MRR
Mean performance	6213.54	207.46	3429.65	128.62	0.0473	0.0473	0.5574
<i>p</i> -spectrum_kernel_SVM + ONBIRES with Original templates	3028	150	3001	148	0.0495	0.0493	0.3740
<i>p</i> -spectrum_kernel_SVM + ONBIRES with Curated Templates	2249	127	2231	126	0.0565	0.0565	0.3696
<i>p</i> -spectrum_kernel_SVM + <i>profile-feature</i> _SVM	5448	352	3210	191	0.0646	0.0595	0.3392

3 Discussion

The protein-protein interaction task of BioCreAtIVe Challenge 2006 has made great strides toward mining *biologically meaningful knowledge* from literature. In our conclusion, there are three aspects that are greatly worth noting: first, identifying physical interactions which will be important for understanding biological functions and processes; second, recognizing protein molecules not from computer scientist's perspectives, but from biologist's perspective (not only identifying entities, but also mapping entities to molecules); third, providing as much biological information as possible for biologists, for instance, experiment detection

method and summary sentences. We call such information as *biologically meaningful knowledge*. In our opinion, this should always be the focus of text mining research in the biology domain.

In this paper, we have presented algorithms and ideas toward mining physical protein-protein information by exploiting abundant features. These features include string, unigram, template-feature and profile-feature. They reflect different angles on the objects to be classified. Features can be selected by statistics, prior knowledge, or other means of computation. Obviously, such an abundant representation is helpful in these tasks.

The most challenging problem in these tasks, in our opinion, is to recognize named entities. NER here is slightly different from traditional NER because we need map entities to molecule identifiers. A better NER module will improve the performance of interaction identification module markedly. Moreover, full texts are much more difficult to handle since there are more inconsistent terminologies, and even more domain-specific knowledge. A breakthrough in NER will benefit all tasks of text mining in the biology domain.

Acknowledgements

We would like to thank Zhiqiang Gu for his implementation of p-spectrum kernel using trie-structure, which had reduced a large amount of computation. We also want to thank Martin Krallinger and his colleagues for organizing such a competition, and also for answering many questions.

The work was supported by Natural Science Foundation of China under grant No. 60572084, and China 863 Program under No. 2006AA02Z321.

References

- [1] Hermjakob, H., Montecchi-Palazzi, L., Lewington, C., Mudali, S., Kerrien, S., Orchard, S., Vingron, M., Roechert, B., Roepstorff, P., Valencia, A., Margalit, H., Armstrong, J., Bairoch, A., Cesareni, G., Sherman, D., Apweiler, R. (2004) IntAct: an open source molecular interaction database. Nucleic Acids Res., 32, D452-D455.
- [2] Zanzoni, A., Montecchi-Palazzi, L., Quondam, M., Ausiello, G., Helmer-Citterich, M., Cesareni, G. (2002) MINT: a Molecular INTeraction database. FEBS Lett., 513, 135-140.
- [3] Marcotte, E.M., Xenarios, I., Eisenberg, D. (2001) Mining literature for protein-protein interactions. Bioinformatics, 17, 259--363.
- [4] Ono, T., Hishigaki, H., Tanigami, A. and Takagi, T. (2001) Automated extraction of information on protein-protein interactions from the biological literature. Bioinformatics, 17, 155-161.
- [5] Huang, M., Zhu, X.Y., Payan, D.G., Qu, K., Li, M. (2004) Discovering patterns to extract protein-protein interactions from full texts. Bioinformatics, 20, 3604-3612.
- [6] http://biocreative.sourceforge.net/biocreative_2_ppi.html
- [7] <http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=MI>
- [8] <http://www.chem.qmul.ac.uk/iubmb/enzyme/>
- [9] <http://htmlparser.sourceforge.net/>
- [10] <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Taxonomy>
- [11] Huang, M.L., Zhu, X.Y., Ding, S.L., Yu, H., Li, M. ONBIRES: Ontology-based biological relation extraction system. 4th Asia-Pacific Bioinformatics Conference, pp. 327-336, FEB 13-16, 2006.
- [12] Ding, S.L., Huang, M.L., and Zhu, X.Y. Semi-supervised Pattern Learning for Extracting Relations from Bioscience Texts. In Proceedings of the 5th Asia-Pacific Bioinformatics Conference, pp. 307-316, Jan 15-17, 2007.



Uncovering Protein-Protein Interactions in the Bibliome

Alaa Abi-Haidar^{1,6}, Jasleen Kaur¹, Ana Maguitman², Predrag Radivojac¹,
Andreas Retchsteiner³, Karin Verspoor⁴, Zhiping Wang⁵, Luis M. Rocha^{1,6 *}

*To whom correspondence should be addressed: rocha@indiana.edu

¹ School of Informatics, Indiana University, USA

² Dep. de Ciencias e Ing. de la Computación, Universidad Nacional del Sur, Argentina

³ Center for Genomics and Bioinformatics, Indiana University, USA

⁴ Information Sciences Group, Los Alamos National Laboratory, USA

⁵ Biostatistics, School of Medicine, Indiana University, USA

⁶ FLAD Computational Biology Collaboratorium, Instituto Gulbenkian de Ciência, Portugal

Abstract

We participated in three of the Protein-Protein Interaction (PPI)subtasks: Protein Interaction Article Sub-task 1 (IAS), Protein Interaction Pairs Sub-task 2 (IPS), and Protein Interaction Sentences Sub-task 3 (ISS). Our approach includes a feature detection method based on a spam-detection algorithm. For IAS we submitted three runs from distinct classification methods: the novel Variable Threshold Protein Mention Model, Support Vector Machines, and an integration method based on measures of uncertainty and a nearest neighbor predictor on the eigenvector space obtained via the Singular Value Decomposition of the feature/abstract matrix. For IPS and ISS we used the features discovered from IAS abstracts as well as features from training IPS and ISS data to predict appropriate passages and pairs. We also used the number of protein mentions in a passage as a feature.

Keywords: Protein interaction, text mining, bibliography informatics, support vector machines, singular value decomposition, spam detection, uncertainty measures, proximity graphs, complex networks.

1 Protein Interaction Article Sub-Task 1 (IAS)

1.1 Feature Selection

All three runs submitted use word features extracted from the training data using a method inspired by the spam filtering system *SpamHunting* (Fdez-Riverola et al., 2007). First, we computed the probability that a word w appears on a positive $p_{TP}(w)$ abstract, as the ratio of the number of positive abstracts containing w , over the total number of positive abstracts. Similarly, we computed the probability that a word w appears on a negative abstract $p_{TN}(w)$. After stemming with the Porter algorithm, filtering out short words with 2 or less letters, and removing common stop words except the word "with", we ranked all words according to the score: $S(w) = |p_{TP}(w) - p_{TN}(w)|$. The words with the highest score S tend to be associated either with positive or negative abstracts. Therefore, such words are assumed to be good features for classification.

1.1.1 Single Word Feature Set

The first feature set we used were the top 650 stemmed abstract words with largest S ; the top 15 words are listed in table 1 in the supplemental materials (section 3 and online ¹), which also includes figure 3 depicting the 1000 abstract words with largest S in the space of p_{TP} and p_{TN} .

¹<http://informatics.indiana.edu/rocha/bc2>

1.1.2 Word Pair Feature Sets

We produced two additional feature sets comprised of word pairs obtained from the 650 stemmed word features in the first set. This leads to $650^2 = 422500$ possible word pairs, though not all occur. First, we removed all words not in the first feature set from the abstracts. Then, from the filtered abstracts we obtain the second and third feature sets, which are comprised of pairs of words immediately adjacent (bigrams) and pairs of words that occur within a window of ten words from each other, respectively. We also computed the probability that such word pairs (w_i, w_j) appear in a positive or negative abstract: $p_{TP}(w_i, w_j)$ and $p_{TN}(w_i, w_j)$, respectively. Figure 4 depicts the 1800 word pairs of the third feature set with largest: $S^{10}(w_i, w_j) = |p_{TP}(w_i, w_j) - p_{TN}(w_i, w_j)|$. Table 1 in the supplemental materials (section 3) lists the top 15 word pairs for S^{10} .

1.1.3 Number of Protein Mentions

Using *A Biomedical Named Entity Recognizer* (ABNER)² (Settles, 2005), we extracted unique protein mentions from abstracts. These mentions were later converted to UniProt IDs only for the IPS and ISS tasks (see section 2); for the IAS task we used the number of unique ABNER protein mentions per abstract a , $np(a)$, as an additional feature or parameter.

1.2 Classification Methods

To test the various classification methods described below, we performed k-fold tests on the supplied training data, as well as additional data from MIPS (positives) and abstracts curated by hand (negatives) that were graciously donated to our team by Santiago Schnell. Based on the results of these tests, we submitted the three runs described below.

1.2.1 Support Vector Machine Classification

Starting from the first feature set (single words with largest S) we applied additional dimensionality reduction and then trained classification models to discriminate between positive and negative data. Dimensionality reduction involved a two-step process. First, a feature selection filter based on the t-test was used in which all features with the p-value below a pre-specified threshold t_f were retained. Then, we applied the principal component analysis (Wall et al., 2003) to retain all features containing $t_{PCA} \cdot \sigma^2$ of the total variance σ^2 . The remaining features were fed into a support vector machine, a classification model used to maximize the margin of separation between positive and negative examples (Vapnik, 1998). We used the *SVMlight* package (Joachims, 2002) in which we explored both polynomial and Gaussian kernels with various parameters. The overall system was trained to maximize the classification accuracy on the unlabeled data using the following two-step iterative procedure: (i) train a classifier with costs adjusted to the current estimates of class priors in the unlabeled data; and (ii) predict class labels on the unlabeled set using current classifier and make new estimates of the class priors. Initially, class priors in the unlabeled data were set to 0.5. Not more than five rounds were executed, ending with the total cost of positive examples being about 3 times the costs of the negatives. The final predictor used $t_f = 0.1$ for the feature filtering, $t_{PCA} = 0.95$ for the principal component analysis and a linear support vector machine.

1.2.2 Variable Trigonometric Threshold Classification

We developed trigonometric measures of term relevance on the p_{TP}/p_{TN} plane. It is obvious that the best features in this plane are the ones that are closest to either one of the axis. Any feature w is a vector in this plane (see figure 1); let α be the angle of this vector with the p_{TP} axis. We then

²<http://www.cs.wisc.edu/~settles/abner/>

use $\cos(\alpha)$ as a measure of feature terms³ mostly associated with positive abstracts, and $\sin(\alpha)$ of terms mostly associated with negative ones (in the training data set). Then, for every abstract a , we compute the sum of all feature term contributions for a positive (P) and negative (N) decision:

$$P(a) = \sum_{w \in a} \cos(\alpha(w)), \quad N(a) = \sum_{w \in a} \sin(\alpha(w)) \quad (1)$$

The decision of whether abstract a is a positive or negative abstract (in so far as being relevant for protein-protein interaction) is then computed as:

$$\begin{cases} a \in TP, & \text{if } \frac{P(a)}{N(a)} \geq \lambda_0 + \frac{\beta - np(a)}{\beta} \\ a \in TN, & \text{otherwise} \end{cases} \quad (2)$$

where λ_0 is a constant threshold for deciding whether an abstract is positive (relevant) or not. This threshold is subsequently adjusted for each abstract a with the factor $(\beta - np(a))/\beta$, where β is another constant, and $np(a)$ is the number of protein mentions in abstract a as described in section 1.1.3. We observed that abstracts have a higher chance of being positive (relevant) with more protein mentions, thus, via formula 2, the classification threshold is linearly decreased as np increases. This means that with a high (lower) number of protein mentions, it is easier to classify an abstract as positive (negative). When $np(a) = \beta$ the threshold is simply λ_0 . We refer to this classification method as *Variable Trigonometric Threshold* (VTT).

The specific value of λ_0 was determined by optimizing the F-Score measure⁴ on the training data as well as on the additional abstracts obtained from MIPS and hand curated. To decide on the best β we computed the probability that a positive abstract a in the training set contains more than np protein mentions: $pos = p(TP|np)$. We also computed the negative counterpart: $neg = p(TN|np)$. We observed that when $np \geq 7$, we maximize $pos - neg$, thus we set $\beta = 7$. This way, when $np(a) > 7$ the decision threshold is lowered, and raised otherwise. Figures 5 and 6 in the supplemental materials (section 3) depict this study. Finally, the run we submitted with VTT used the following parameters: $\lambda_0 = 1.7$ and $\beta = 7$, using the top 650 word-pair features of the third feature set (section 1.1.2). This combination of parameters resulted in the best F-Score values for the training and additional data.

1.2.3 Classification with Singular Value Decomposition Plus Uncertainty Integration

We first classified the set of abstracts using a nearest neighbor classifier on the eigenvector space obtained via the *Singular Value Decomposition* (SVD) (Wall et al., 2003) of the feature/abstract space. We used the first feature set of single word features (section 1.1.1). We represented abstracts as vectors in this feature space. We then calculated the inverse document frequency measure (IDF), so the vector coefficients were the TF*IDF (Dumais, 1990) for the respective features. The number of protein mentions per abstract a , $np(a)$ (section 1.1.3), was added as an additional feature. The abstract vectors were also normalized to Euclidean length 1. We computed the SVD of the resulting abstract-feature

³By term, we refer to features in our three different feature sets as described in section 1.1.

⁴F-measure is defined as $F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$, where *Precision* is the proportion of abstracts returned that are relevant (positive), and *Recall* is the proportion of relevant abstracts that are retrieved.

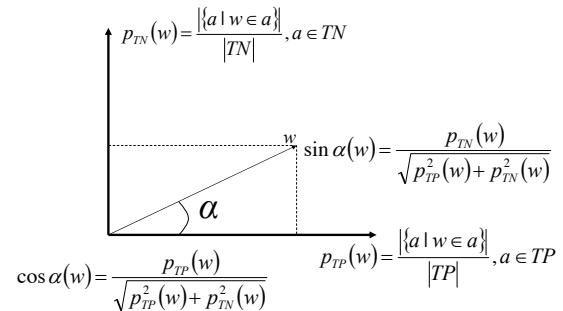


Figure 1: Trigonometric measures of term relevance for identifying positive and negative abstracts in the P_{TP} and P_{TN} plane.

matrix (from the training data). The top 100 components were retained (this number provided best results on our tests on training and additional data). To classify a test abstract vector a , we project it onto this SVD subspace and calculate the cosine similarity measure of a to every training abstract t :

$$\cos(a, t) = \frac{a \cdot t}{\|a\| \times \|t\|} \quad (3)$$

We then calculate positive and negative scores for each test abstract a by summing the cosine measure for every positive ($t \in TP$) and negative ($t \in TN$) training abstract, respectively:

$$P(a) = \frac{1}{|TP|} \sum_{t \in TP} \cos(a, t), \quad N(a) = \frac{1}{|TN|} \sum_{t \in TN} \cos(a, t) \quad (4)$$

where $|TP|$ and $|TN|$ are the number of positive and negative abstracts in the training data, respectively. Finally, a linear decision boundary was determined in the two-dimensional space of P and N : abstract a is classified as positive (relevant) if $P(a) > m \cdot N(a) + b$ and as negative otherwise. Coefficients m and b were determined manually from optimizing the F-Score measure on the training data as well as on the additional abstracts obtained from MIPS and hand curated. Figure 7 in the supplemental materials (section 3) depicts the boundary surface in the P and N space.

Using a variation of a method we previously used (Maguitman et al., 2006), we integrated three variations of the VTT classification (section 1.2.2) with the SVD classification in such a way that for each abstract the most “reliable” prediction was used to issue a classification. To ascertain reliability, we represented the target test abstract a , as well as all abstracts t in the training data, as vectors in a compound feature space (including all three feature sets described in section 1.1). Next, we computed the cosine similarity, $\cos(a, t)$, between a target a and every t , and treated this value as a weighted vote. Thus, if abstract t is very close to a , it will have a greater influence in the classification of a . Since for any abstract t in the training data, we know if a classification method correctly classified it, we tried two different measures of reliability:

- **Entropy-Based Measure:** As in (Maguitman et al., 2006), we used Shannon’s entropy to compute the uncertainty of a prediction for the target abstract based on the distribution of positive and negative weighted votes obtained for that abstract from a given classification method.

Let $\rho_M(a, TP)$ and $\rho_M(a, TN)$ be the probabilities of predicting TP or TN , respectively, as the class for abstract a using method M . We estimate these probabilities as follows:

$$\rho_M(a, TP) = \frac{\sum_{t \in TP} \cos(a, t)}{\sum_{t \in TP \cup TN} \cos(a, t)}, \quad \rho_M(a, TN) = \frac{\sum_{t \in TN} \cos(a, t)}{\sum_{t \in TP \cup TN} \cos(a, t)}.$$

Note that $\rho_M(a, TP) = 1 - \rho_M(a, TN)$. Finally, we compute the prediction uncertainty of abstract a using method M , $U_M(a)$, using Shannon’s entropy as follows:

$$U_M(a) = -\rho_M(a, TP) \log \rho_M(a, TP) - \rho_M(a, TN) \log \rho_M(a, TN)$$

Using the uncertainty measure we integrate the predictions issued by each method by selecting, for each abstract a , the prediction issued by the method M with lowest $U_M(a)$

- **Misprediction Measure:** We used the information about correct predictions available from the training set to compute a *misprediction* rate from each classification method; each neighbor t of the target abstract a contributed to a method’s rate based on its weighted vote.

Assume T is the training set of abstracts, and $I^M \subseteq T$ be the set of abstracts that has been misclassified using method M . Let $\mu_M(a)$ be the misprediction rate for abstract a based on the weighted votes for a from abstracts $t \in I^M$:

$$\mu_M(a) = \frac{\sum_{t \in I^M} \cos(a, t)}{\sum_{t \in T} \cos(a, t)}$$

Using this misprediction rate we integrate the predictions issued by each method by selecting, for each abstract a , the prediction issued by the method M with lowest $\mu_M(a)$

Finally, we calculated the product of the Entropy-Based Measure and the misprediction Measure and selected, for each target test abstract a , the prediction coming from the classification method with lowest product. In our submission for run 3, we used this uncertainty-driven integration with the following four classification methods:

1. SVD Vector model with first feature set of single words.
2. Fixed Threshold Classification (FT). This is the same as the VTT classification (section 1.2.2) but without trigonometric measures. In this case, instead of the formulae 1, we simply used:

$$P(a) = \sum_{w \in a} p_{TP}(w), \quad N(a) = \sum_{w \in a} p_{TN}(w) \quad (5)$$

We also did not use the ABNER protein mention counts, thus formula 2 becomes simply $P(a)/N(a) > \lambda_0 = 1.3$. In this case, we also used the first feature set of single words.

3. VTT exactly as described in section 1.2.2, but with the second feature set (bigrams) and $\lambda_0 = 1.5$ and $\beta = 7$.
4. VTT exactly as described in section 1.2.2.

Items 2 to 4 were chosen so that there would be a model for each of the three feature sets. The specific parameters were chosen from the F-score performance with the learning and additional data. It is important to notice that in our tests, the uncertainty-driven integration algorithm (SVD-UI) improved only very slightly over the SVD vector model alone. Indeed, for the test set the SVD vector model alone produced the same classification as the integration method, except that different rankings of abstracts were attained. We decided to submit the results of the integration method because it slightly improved on the SVD vector model with the learning data.

1.3 Results

The performance of the three runs we submitted (sections 1.2.1, 1.2.2, and 1.2.3) can be seen in Table 2 of the supplemental materials (section 3). The three runs produced similar results regarding the F1 measure (*F-Score*), with the highest value (0.75) for Run 2 (VTT, section 1.2.2), and lowest (0.73) for Run 3 (SVD-UI, section 1.2.3). However, this measure hides the distinct capabilities of each method. Indeed, the SVM method resulted in the best recall and worst precision (0.88/0.64), whereas the VTT method resulted in the worst recall and best precision (0.79/0.71). The SVD-UI method lies in between the other two (0.8/0.68), though its F-Score measure is slightly worse.

Perhaps a better measure for this task is *accuracy*, which gives us the ratio of correct predictions (for both positive and negative abstracts). In this case, the VTT method yielded the best result (0.74),

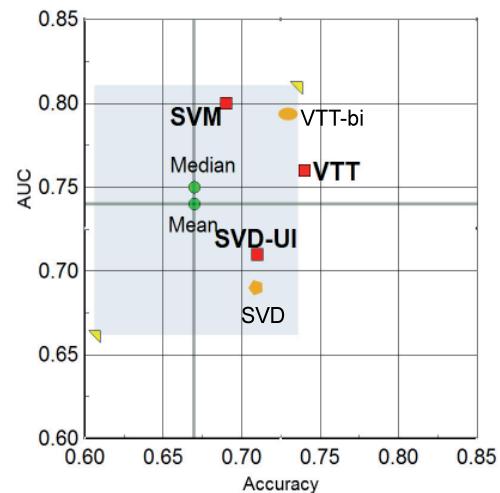


Figure 2: Our methods on the accuracy vs. AUC plane for IAS. Mean and Median are for the set of all submissions from all groups. Red squares denote our three submissions (SVM, VTT, and SVD-UI). The orange polygon denotes the results for SVD alone, and the orange oval denotes the results for one of the versions of VTT (with bigrams) included in the SVD-UI method.

followed by the SVD-UI (0.71), and the SVM (0.69) methods. Thus, the VTT method lead to a more balanced prediction for both positive and negative abstracts, leading to the lowest error rate (0.26).

When we look at the *Area Under the ROC Curve* (AUC) measure, however, the results have yet another reading. This measure can be understood as the probability that for a randomly picked positive abstract and a randomly picked negative abstract, the positive abstract is ranked above the negative one. We obtained very good results with this measure for the SVM run (0.8), followed by good results for the VTT (0.76) and SVD-UI (0.71) methods. Contrasting accuracy with AUC, we observe that while the VTT method lead to our highest accuracy, the probability of finding a false positive closer to the top of the rank (or a false negative closer to the bottom of the rank) is larger than in the ranking produced by the SVM method (see figure 8). This situation was even worse with the SVD-UI method, as can be seen in figure 9, where many negative (positive) abstracts appear deep in the positive (negative) side of the decision surface. Conversely, while the SVM method lead to our lowest accuracy measure, it yielded the highest AUC, which indicates that a larger proportion of its erroneous decisions were closer to its decision surface.

As we discuss in section 1.2.3, the SVD vector model alone produced the same classification as SVD-UI, except that different rankings of abstracts were attained. We note that the AUC of the SVD method alone was lower³(0.68) than that of the AUC-UI method (0.71). We can thus say that the integration method improved the AUC of the SVD method alone. However, it produced worse AUC and accuracy values for other constituent methods, such as VTT as submitted in Run 2. Indeed, the AUC and accuracy of the not submitted individual methods included in the uncertainty integration method (section 1.2.3), show that constituent method 3, a version of VTT method using bigrams, produces a higher AUC (0.79) than the VTT we submitted and the SVD-UI method, without sacrificing accuracy much (0.73). This means that the integration method did worse than some of its constituents, and that the VTT method can produce better results. A comparison of all our methods in the AUC/Accuracy plane is depicted in Figure 2. The figure also contrasts our results with the central tendency of all group submissions. The most salient points are:

- **Accuracy:** All three runs are above the mean and median values of accuracy for all teams. Run 2 (VTT) yielded an accuracy above one standard deviation of the mean accuracy.
- **AUC:** Both the SVM and VTT methods are above the mean and median value of AUC, but the SVM method is very nearly above one standard deviation above the mean.
- **Balance across all performance measures:** The VTT method was the only one which was above the mean for all measures tested (precision, recall, F-score, accuracy, and AUC).

2 Protein Interaction Pairs And Sentences Sub-Tasks (IPS AND ISS)

2.1 Feature Selection

From the IAS subtask, we collected the top 1000 word-pair features, (w_i, w_j) from the third feature set (section 1.1.2). Since the purpose of these tasks is to identify portions of text where protein-protein-interaction (PPI) information appears, we do not need to worry about features indicative of negative PPI information. Features are chosen and ranked according to high value of:

$$p(w_i, w_j) = p_{TP}(w_i, w_j) \cdot \cos(\alpha(w_i, w_j)) = \frac{p_{TP}^2(w_i, w_j)}{\sqrt{p_{TP}^2(w_i, w_j) + p_{TN}^2(w_i, w_j)}} \quad (6)$$

where p_{TP} and p_{TN} are as defined in section 1.1. We multiply the cosine measure by the probability of the feature being associated with a positive abstract, to ensure that features which have zero probability of being associated with a negative abstract (there are many), are not equally ranked with $p(w_i, w_j) = 1$. We refer to this set of 1000 stemmed word pairs, as the *word pair feature set*.

We also obtained an additional set of features from PPI-relevant sentences: the *sentence feature set*. These sentences were extracted from the various files provided by Biocreative for these tasks. We collected all sentences that contained PPI, and calculated the frequency of stemmed words in this collection: $f_{ppi}(w)$. Then we calculated the frequency of stemmed words of the entire corpus: $f_c(w)$. Finally, similarly to the word pair features above, we selected as sentence features the top 200 words which maximize the following score (top 10 listed in Table 3.):

$$SS = \frac{f_{ppi}^2(w)}{\sqrt{f_{ppi}^2(w) + f_c^2(w)}} \quad (7)$$

2.2 Paragraph Selection and Ranking

We used our two feature sets plus protein mention information to select the paragraphs in each document which are more likely to contain PPI information. Thus, for each full text document, we ranked paragraphs according to three different criteria:

- A Largest sum of word pair feature weights (section 2.1), where the weights are the inverse feature rank. Paragraphs without feature matches are thrown out (rank 0).
- B Largest number of protein mentions in paragraph. As in the IAS subtask (see section 1.1.3), we also used ABNER to collect protein mentions in the full text documents provided for these two subtasks. Paragraphs without protein mentions are thrown out (rank 0).
- C Largest number of sentence features in paragraph (section 2.1). Each feature that occurs in a paragraph adds 1 to the count. Paragraphs without feature matches are thrown out (rank 0).

From these three distinct paragraph rankings, for each document, we produced another three rankings that aim to integrate this information in different ways. For each document, we rank paragraphs according to the following criteria:

1. Rank product of ranks produced in A (word pair features) and B (protein mentions) above.
2. Rank product of ranks produced in B (protein mentions) and C (sentence features) above.
3. Rank product of ranks produced in A, B, and C above.

Since paragraphs thrown out in A, B and C are rank 0, in this step, only paragraphs with feature matches and protein mentions remain. The resulting 3 rankings constitute the paragraph rankings in the three runs submitted for the IPS subtask: 1, 2, and 3, respectively.

2.3 Mapping of Protein Mentions to UniProt IDs

To obtain the actual protein-protein interaction pairs contained in the paragraphs of ranks 1, 2, and 3 described in section 2.3, we had to convert the textual mentions obtained with ABNER to UniProt IDs. Protein and gene references identified using the ABNER system were mapped to UniProt IDs through exact matching with either a gene or a protein name occurring in SwissProt—considering both primary names and synonyms. UniProt version 8.2 was used for the mapping; this is not the most current version and could have resulted in missing relevant mappings. These mappings were then filtered using the provided UniProt subset. This process typically resulted in many UniProt IDs for the same ABNER protein mention, mostly because the same protein name maps to different UniProt IDs for different organisms. We therefore filtered the protein mention to include only UniProt ID mappings associated with organisms in the set of MeSH terms of a given article. Unfortunately, many of the articles listed several organisms in their MeSH keyterms. An obvious improvement would be to detect the appropriate organism for a given paragraph more specifically.

2.4 Selection and Ranking of Protein-Protein Interaction Pairs for IPS

Finally, for the IPS task, we returned all the combinations of protein pairs (UniProt accession numbers as described in section 2.3) occurring in the same sentence—for sentences included in the paragraphs of ranks 1,2, and 3 (section 2.2). For a given document (PMID), the rank of each protein-protein interaction pair is the rank of the highest ranked paragraph in which the pair occurs in a sentence. We submitted three distinct rankings of PPI pairs according to the three ranks 1,2, and 3 (section 2.2).

2.5 Protein Mention Feature Expansion with Proximity Networks

We used a method we employed in the first Biocreative competition to obtain additional, contextualized features associated with a protein names (Verspoor et al., 2005), that is, additional features which are relevant in a specific document, but not necessarily in the whole corpus. We computed for each document a word proximity network based on a co-occurrence proximity measure of stemmed words in paragraphs of that document:

$$WPP(w_i, w_j) = \frac{\sum_{k=1}^m (r_{i,j} \wedge r_{i,j})}{\sum_{k=1}^m (r_{i,j} \vee r_{i,j})} \quad (8)$$

where $r_{i,j} \in \{0, 1\}$ is an element of the relation $R : P \times W$; P is the set of all m paragraphs in a document, and W is the set of all n stemmed words. This yields a proximity network for each document, where the nodes are words w_i , and the edges are the $WPP(w_i, w_j)$ proximity weights.

Next, for every PPI pair (obtained by rank 1) occurring in a given document, we obtain the words closest to the protein labels in the network. Notice that these protein labels are words identified by ABNER for the given PPI pair, and they appear on the proximity network as regular nodes. For each protein pair we selected the 5 stemmed words (nodes) in the proximity network with largest minimum proximity to both protein names. The sentences in the articles where the PPI pairs occur were then augmented using the 5 words obtained from the relevant document.

2.6 Passage Extraction and ISS Submission

From ranked paragraphs, we selected passages (sets of 3 sentences) containing a given PPI pair. Finally, we submitted three runs to the ISS subtask:

1. Passages ranked by largest number of occurring word pair features (see section 2.1).
2. Passages ranked by largest number of occurring word pair features, but where the PPI occurring sentence is expanded with words from the document’s proximity network.
3. Same as 2, with the addition of the following factor $100/\text{paragraph_rank_1}$ (see section 2.2) to the number of features found in the passage.

2.7 Results

The results for the IPS and ISS tasks were disappointing, though in line with the central tendency of all submissions. Our three submitted runs to IPS were hardly distinguishable. For all our three runs, the precision was below the mean and median of all submissions, but still well within one standard deviation of the mean. On the other hand, recall was above the mean and median of all submissions, and above one standard deviation of the mean. The F-score was very near the mean and median of all submissions. These results were true for both the identification of protein-protein interaction pairs (PPIN) and single interactors (PN), as well as for the set of all articles (All) and the subset of articles

containing exclusively SwissProt interaction pairs (SP). Figure ?? in the supplemental materials lists the specific values.

Regarding the ISS subtask, the three submitted runs were slightly different, and denoted a slight improvement with the number of the run. Run 2 was better than run 1, which shows that the proximity expansion improved a little the original features. Run 3 was better than run 2, showing that considering the paragraph rank from IPS (which includes number of protein mentions) in addition to the expanded word-pair features is advantageous. Again our results were in line with the averaged values of all submissions. Our matches (387) and unique matches (156) to previously selected passages were above the average of all submissions (207.46 and 128.62, respectively). We should notice, however, that we predicted many more passages (18371) and unique passages (5252) than the average (6213.54 and 3429.65, respectively), which lead to lower than average fractions of correct from predicted and unique passages. Like in the IPS case, this means that our system was better at recall than at precision. Finally, our mean reciprocal rank of correct passages substantially higher than average (0.66 to 0.56). Table 4 in the supplemental materials lists the specific values.

Acknowledgements

We would like to thank Santiago Schnell for graciously providing us with additional proteomics related articles not containing protein-protein interaction information. We would also like to thank the FLAD Computational Biology Collaboratorium at the Gulbenkian Institute in Oeiras, Portugal, for hosting and providing facilities used to conduct part of this research during the summer of 2006. We are grateful to Indiana University's Research and Technical Services for technical support. The AVIDD Linux Clusters used in our analysis are funded in part by NSF Grant CDA-9601632.

References

- Dumais, S. (1990). Enhancing performance in latent semantic indexing. Technical Report TM-ARH-017527, Bellcore.

Fdez-Riverola, F., Iglesias, E., Diaz, F., Mendez, J., and Corchado, J. (2007). Spambunting: An instance-based reasoning system for spam labelling and filtering. *Decision Support Systems*, In Press.

Joachims, T. (2002). *Learning to classify text using support vector machines: methods, theory, and algorithms*. Kluwer Academic Publishers.

Maguitman, A., Rechtsteiner, A., Verspoor, K., Strauss, C., and Rocha, L. (2006). Large-scale testing of biolome informatics using pfam protein families. In *Pacific Symposium on Biocomputing*, Vol. 11, pages 76–87.

Settles, B. (2005). Abner: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192.

Vapnik, V. (1998). *Statistical learning theory*. John Wiley & Sons, New York.

Verspoor, K., Cohn, J., Joslyn, C., Mniszewski, S., Rechtsteiner, A., Rocha, L., and Simas, T. (2005). Protein annotation as term categorization in the gene ontology using word proximity networks. *BMC Bioinformatics*, 6 Suppl 1:S20.

Wall, M., Rechtsteiner, A., and Rocha, L. (2003). Singular value decomposition and principal component analysis. In Berrar, D., Dubitzky, W., and Granzow, M., editors, *A Practical Approach to Microarray Data Analysis*, pages 91–109. Kluwer, Norwell, MA.



An integrated approach to concept recognition in biomedical text

William A. Baumgartner, Jr.¹ Zhiyong Lu¹ Helen L. Johnson¹
J. Gregory Caporaso¹ Jesse Paquette¹ Anna Lindemann
Elizabeth K. White Olga Medvedeva K. Bretonnel Cohen¹
Lawrence Hunter¹

¹ Center for Computational Pharmacology, University of Colorado School of Medicine

Abstract

Our approach to the three BioCreative 2006 tasks had three main characteristics: (1) Extensive use of UIMA (Unstructured Information Management Architecture), a framework that facilitates integration and evaluation of system components, as well as incorporation of third-party tools. (2) Extensive use of a semantic parser, OpenDMAP (Open Source Direct Memory Access Parser). (3) Use of domain-specific rule-based approaches for handling coordination of protein names. We noted large differences between our performance on the training data and our performance on the test data in the IAS and IPS sub-tasks.

Keywords: semantic parsing, conceptual language processing, knowledge-based language processing, direct memory access parsing (DMAP)

1 Introduction

The approach of the Center for Computational Pharmacology to the BioCreative 2006 tasks had three basic characteristics: (1) use of an architecture that allowed us to apply a single, integrated framework to all three tasks; (2) extensive use of a semantic parser; and (3) use of rule-based approaches to handling coordination of protein names.

We made extensive use of the UIMA (Unstructured Information Management Architecture) [11, 20] framework for integrating almost every component that we used in any BioCreative 2007 task. Three benefits accrued from this strategy: (a) The complete integration of all processing steps allowed us to quickly and easily experiment with different approaches to the many subtasks involved. (b) It made it easy for us to quickly evaluate the results of these experiments against the official data sets. (c) It provided us with a clean interface for incorporating tools from other groups, including LingPipe [4], ABNER [28], and Schwartz and Hearst's [27] abbreviation detection algorithm.

We also made extensive use of a semantic parser being developed by our group. Called OpenDMAP (Open source Direct Memory Access Parser), it is a modern implementation of the DMAP paradigm first developed by Riesbeck [26], Martin [21], and Fitzgerald [12]. The earliest descriptions of the paradigm assumed that a DMAP system would approach all levels of linguistic analysis, from part-of-speech choice through word sense discrimination to extraction of propositional content, through a single optimization procedure. In this work, we show that analysis can be modularized, and even externalized, without losing the essential semantic flavor of the DMAP paradigm.

Finally, we developed a number of rules for handling the domain-specific conjunction strategies of biomedical text, such as using *BMP1/2* to mean “BMP 1 and BMP 2.”

2 Gene Mention Task

Our system for the 2006 Gene Mention (GM) task focuses on simple consensus approaches for combining the output of multiple gene taggers. We used three taggers: an in-house tagger developed for the BioCreative 2004 gene mention task (Task 1A) [16] and two publicly available taggers, ABNER

[28] and LingPipe [4]. Integration of each tagger into the system was accomplished using the UIMA [11, 20]. Our overall consensus approach can be divided into two general strategies which test two distinct hypotheses. *Hypothesis #1* poses that filtering the output of multiple gene/protein mention identification systems by requiring agreement by two or more of the individual systems will result in a precision measure greater than or equal to the highest precision measure of the individual components. *Hypothesis #2* states that combining the output of multiple gene/protein mention identification systems will result in a recall measure greater than or equal to the highest individual recall measure of the individual components.

To test these hypotheses, we implemented two filter varieties which combine the output of gene taggers in different ways. It should be noted that the taggers used for the GM task were used “out-of-the-box,” that is, they were not trained on the BioCreative 2006 data. The models used for each tagger were trained on data from the inaugural BioCreative gene mention task, and judging from the results, each tagger was trained on different parts of the original data (Table 1).

Consensus Filter: To test Hypothesis #1, we developed a consensus filter, analogous to a voting scheme. Each tagger votes, and a gene mention is kept if it accumulates a certain threshold of votes. If the threshold is not met, the gene mention is removed from the analysis. We used two consensus approaches, one which required two of the three taggers to agree, and the other which required unanimous agreement in order for a gene mention to be kept. By combining the output of three taggers which are known to have decent performance on their own, we expected that the consensus approach would result in an elevation in overall precision, without dramatically decreasing recall. Although it might seem intuitively sensible to weight the vote of each tagger, perhaps by the performance data reported in Table 1, that data is not clearly directly comparable, since each tagger was trained on different subsets of the 2004 data. Therefore, we weighted each tagger’s vote equally.

Overlapping Filter: To maximize recall (and test Hypothesis #2), we implemented a simple filter which keeps all gene mentions by resolving overlapping mentions among the taggers. When an overlap between two gene mentions is detected, the filter compares their respective span lengths, and keeps the gene mention with the greater span¹. By keeping all gene mentions, we expected to increase the recall of the system; however, we also expected the precision of the system to suffer, since more false positives will be returned.

Table 1: Performance of the individual gene taggers on the 2006 training data broken down according to the 2004 BioCreative data sets.

Tagger	2004 Test			2004 Train			2004 Dev		
	P	R	F	P	R	F	P	R	F
CCP	77.5	77.9	77.7	88.5	88.6	88.6	81.2	79.3	80.2
ABNER	78.0	73.7	75.8	89.2	89.0	89.1	78.0	70.7	74.2
LingPipe	88.1	92.6	90.3	88.6	92.9	90.7	88.5	92.5	90.5

Results: We conducted a simple experiment to gauge the differences in training data used for each of the three taggers. Table 1 shows the performance of each tagger on the 2006 data. The data has been divided according to the three different data sets provided in the inaugural gene mention task, *test*, *train* and *dev*. Having constructed the CCP tagger in-house, we know that it was trained on the *train* and *dev* portions of the data which is reflected in the performances depicted in the table. The results of this experiment suggest that our implementation of ABNER was trained on only the *train* data, while the LingPipe model used was generated using all three subsets of the data.

As expected the consensus approaches increased precision over the individual tagger performances for the training data (See Table 2). For the overlapping filter, we actually note a worse recall than for

¹An alternative would be to return the shortest overlapping span; having noted that BioCreative 2004 Task1A systems that took steps to extend multi-word name boundaries rightward and leftward benefitted from doing so, we chose the longer span.

Table 2: Performances of systems and individual components on the 2006 test and training data. Median score, as supplied by organizers. Quartiles for our runs are shown in parentheses.

Tagger	Test Data			Training Data		
	Precision	Recall	F-measure	Precision	Recall	F-measure
CCP	77.30	77.74	77.52	83.68	83.48	83.58
ABNER	80.38	73.26	76.65	83.85	80.86	82.33
LingPipe	72.53	80.00	76.09	88.47	92.77	90.57
2/3 Majority	85.54 (2)	76.83 (3)	80.95 (3)	91.15	86.33	88.68
Unanimous	92.78 (1)	49.12 (4)	64.24 (4)	94.56	61.41	74.46
Overlap	66.22 (4)	83.72 (2)	73.94 (4)	79.27	91.17	84.80
Median	85.08	79.05	81.32			

LingPipe individually, however, since LingPipe appears to have been trained on the entire data set, it is reasonable to expect the overlapping filter to have a lower recall in this case.

The test data provides a more accurate testing ground for our hypotheses. As with the training data, the consensus approaches are observed to elevate precision over any of the individual components. The overlapping filter also behaved as expected, by increasing the system’s overall recall measure, however the dramatic decrease in precision is an unfortunate side effect.

3 Gene Normalization Task

Most aspects of our approach to the GN task are fairly conventional: we identify gene mentions; normalize typographic and morphological variants; look for a unique Entrez Gene entry to map to; if multiple entries are found, disambiguate between them. The primary novelty of our approach lies in the steps that we take to deal with conjunction.

Gene mention localization: In the gene mention localization step, we focussed on maximizing recall, while taking very basic steps to avoid false positives. To maximize recall, we applied six separate GM systems([14],[29],[16],[4], and [28] with the BioCreative 2004 model and the NLPBA model) and the *overlapping filter* described in Section 2 above. After manually examining false positives output by this system when run on the training data, we developed a set of 9 heuristics (listed in Table 3) to filter out obvious false positives and implemented them using regular expressions. Application of all 9 heuristics resulted in removal of 1086 putative gene mentions, and an increase of precision from 0.770 to 0.829 and of recall from 0.673 to 0.725 on the GN task.

Table 3: Effects of distractor string removal rules on GN scores. Step 0 means no preprocessing steps applied. At each step, the preprocessing rules that precede it are also applied. *Removed* refers to the cumulative number of gene mentions removed.

Removal of ...	Example	P	R	F	Removed
0		0.770	0.673	0.718	0
1 gene chromosome location	<i>3p11-3p12.1</i>	0.772	0.673	0.719	34
2 single, short lowercase word	<i>heme</i>	0.778	0.672	0.721	112
3 strings of only numbers &/or punct	<i>9+/-76</i>	0.779	0.672	0.722	206
4 extra preceding words	<i>protein SNF to SNF</i>	0.790	0.681	0.731	225
5 extra trailing words	<i>SNF protein to SNF</i>	0.812	0.723	0.765	419
6 amino acids	<i>Ser-119</i>	0.815	0.723	0.766	460
7 protein families	<i>Bcl-2 family proteins</i>	0.816	0.722	0.766	701
8 protein domains, motifs, fusion	<i>SNH domain</i>	0.828	0.722	0.771	883
9 non-human proteins	<i>rat IFN gamma</i>	0.829	0.725	0.774	1086

Conjunction resolution: We noted that approximately 8% (52/640) of gene names in the development data set contained conjunctions—either General English ones, e.g. *HMG1 and 2*, or domain-specific ones, e.g. *IL3/5*. We developed a procedure for extracting individual gene names from conjunctions of the following types:

1. Gene names in regular coordinated structures (e.g. *IL3/IL5* refers to *IL3* and *IL5*).
2. Individual gene names in a series omitted (e.g. *freac1-freac7* refers to *freac1*, *freac2*, *freac3*, *freac4*, *freac5*, *freac6* and *freac7*).
3. Gene subtypes separated after the main gene name (e.g. *IL3/5* refers to *IL3* and *IL5*).
4. Gene subtypes separated before the main gene name (e.g. *M and B creatine kinase* is transformed to *M creatine kinase* and *B creatine kinase*).

The algorithm first looks for two typical conjunction-indicating words: *and* and *to*—and two atypical, domain-specific conjunction-indicating forms: *forward slash* (/), and *hyphen* (-). Then the algorithm builds the individual gene names from the conjoined structure. (See [19] for further details.)

Table 4 shows the overall improvement in performance on the training data yielded by the conjunction resolution step. It is slight—F-measure increases only from 0.763 to 0.777, even though as we pointed out above, about 8% of gene tokens in the data appear in structures requiring some processing. One reason for this is that some of the conjoined genes are also mentioned individually, allowing for their normalization without having to handle the conjunction. Another reason is that some conjunctions were beyond the scope of our algorithm, e.g. *granulocyte (G-)* and *granulocyte-macrophage (GM-) colony-stimulating factor (CSF)*).

Table 4: GN results on the training data with and without conjunction resolution.

Steps	Precision	Recall	F-measure
without conjunction resolution	0.836	0.691	0.757
with conjunction resolution	0.827	0.727	0.774

Regularization of typographic and morphological variants: We built a dictionary of gene names and symbols, and then used a set of heuristics to regularize all gene mentions in the dictionary and in the output of the GM step.

Dictionary construction: We extracted the gene symbol, synonyms, and full name from Entrez Gene. In addition to Entrez Gene ², we also investigated other databases such as UniProt ³ and a combination of the two databases. We found the Entrez Gene database to be the best resource for gene dictionary construction for the current task (See Table 5). This result is consistent with the conclusions reported in [7].

Examination of the dictionary entries showed that some entries could be removed without adversely affecting system performance because they are of no use for gene normalization tasks. Four pruning rules were implemented to facilitate their removal. Gene entries that begin with “LOC” or were preceded by “similar to” are temporary and often become discontinued in Entrez Gene, and are therefore unlikely to appear in text. Genes that were classified as either “hypothetical” or as a “pseudogene” were also excluded. These four classes of gene names were removed from the dictionary as it has been shown that smaller gene dictionaries have advantages over larger dictionaries [34]. However, it should be noted that removal of these four gene name classes had no impact on system performance. The dictionary used for the GN task contained 21,206 gene entries; see Table 6 for details.

Gene mention regularization: We then used a set of heuristics to regularize all gene names and symbols in the dictionary and in the output of the GM step. These heuristics are based on our

²*Homo_sapiens.agz.gz* file available at <ftp://ftp.ncbi.nih.gov/gene/>

³<http://www.uniprot.org>

Table 5: GN results on the development set using different online resources for lexicon construction.

Resources	Genes Entries	Precision	Recall	F-measure
Entrez Gene	21,206	0.827	0.727	0.774
UniProt	18,580	0.834	0.591	0.692
EG + UniProt	24,182	0.827	0.708	0.762

Table 6: The number of Entrez Gene entries removed in the *Homo_sapiens.agz.gz* file downloaded on 16 August 2006 according to the filtering rules. Rules are applied in order.

	Matching Term	Removed Gene Entries	Remaining Gene Entries
1	<i>LOC\d+</i>	14,831	24,273
2	<i>similar to</i>	86	24,187
3	<i>hypothetical</i>	264	23,923
4	<i>pseudogene</i>	2,717	21,206

own early work and on previous dictionary-based systems [9, 7, 10]. Table 7 shows the effects of the individual rules on performance. In particular, transformation rules for case normalization and space removal played roles in improving recall; the last rule for removing very short strings enhanced precision by quite a large margin. Use of all seven rules, in sequential order, resulted in an increase of F-measure from 0.586 to 0.774.

Mapping mentions to Entrez Gene IDs: After the extracted gene mentions have been regularized, and conjunctions have been addressed, the processed mentions are compared to all entries in the dictionary using exact string matching. If there are multiple matches, then all of the matched entries are taken into the disambiguation step discussed below. After the extracted gene mentions have been normalized, and conjunctions have been addressed, the processed mentions are compared to all entries in the dictionary using exact string matching. Three outcomes are possible:

1. If there is no match, then nothing is returned.
2. If there is a single match, then it is returned.
3. If there are multiple matches, then all of the matched entries are taken into the disambiguation step discussed below.

In addition to exact string matching, we also investigated some approximate string matching techniques. Like [10], we found that approximate matching markedly increased search time but did not markedly improve performance.

Gene Name Disambiguation: For a given species, a gene name is *ambiguous* when it refers to more than one standard database identifier. For example, *CHE1* is used as a synonym for two separate Entrez Gene entries: *CHE1* (GeneID: 8197) and *CDC2L5* (GeneID: 8621). It has been estimated that > 5% of terms for a single organism are ambiguous [32, 5] and that approximately 85% of terms are ambiguous across species. For the (single-species) GN task, we implemented two approaches to gene name disambiguation. The first method attempts to identify “definitions” of gene symbols, using the Schwartz and Hearst algorithm [27]. Our second approach is similar to that of ([17]), except that it uses the five tokens that appear before and after the ambiguous gene, rather than the entire sentence. In both cases, we calculate the Dice coefficient between the extracted text (abbreviation definitions or flanking tokens) and the full name of each gene candidate as given in Entrez Gene. (Our implementation of the Dice coefficient calculation uses a stop word list and stems each token [24]. The gene with the highest non-zero Dice coefficient is returned. If the Dice coefficients are all zero, we return nothing.)

Our results indicate that finding unabbreviated gene names or flanking words plays an important role in resolving ambiguous terms (see Table 8). Moreover, this gene name disambiguation procedure can provide evidence for a term being a false gene mention. For example, *STS* (PMID: 7624774) is

Table 7: Heuristics used to normalize gene names in both lexicon construction and during processing of the gene tagger output, and the results after each step was performed. Step 0 means no string transformation was applied. At each rule, the processing rules that precede it are also applied.

Rule	Example	P	R	F
0		0.783	0.469	0.586
1 Substitution: Roman letters >arabic numerals	<i>carbonic andydrase XI</i> to <i>carbonic andydrase 11</i>	0.778	0.492	0.603
2 Substitution: Greek letters >single letters	<i>AP-2alpha</i> to <i>AP-2a</i>	0.779	0.497	0.607
3 Normalization of case	<i>CAMK2A</i> to <i>camk2a</i>	0.787	0.619	0.693
4 Removal: parenthesized materials	<i>sialyltransferase 1 (beta-galactoside alpha-2,6-sialyltransferase)</i> to <i>sialyltransferase 1</i>	0.782	0.623	0.694
5 Removal: punctuation	<i>VLA-2</i> to <i>VLA2</i>	0.768	0.667	0.714
6 Removal: spaces	<i>calcineurin B</i> to <i>calcineurinB</i>	0.784	0.742	0.762
7 Removal: strings <2 characters	<i>P</i>	0.827	0.727	0.774

Table 8: Results of gene normalization with and without disambiguation.

Steps	Precision	Recall	F-measure
without disambiguation	0.848	0.689	0.760
use abbreviations only	0.825	0.722	0.770
use abbreviations and flanking regions	0.827	0.727	0.774

recognized as a gene mention, but its surrounding words, *content mapping and RH analysis*, indicate it is an experimental method. We assembled a list of words suggesting non-protein terms such as *sequence* or *analysis*. When they were matched to a gene’s unabbreviated name or its flanking words, the gene is considered as a false mention.

Even with the improvement yielded by disambiguation, ambiguity remains a contributor to system error: on the development data, our precision for mentions that only matched a single Entrez entry was 0.85, while for ambiguous entries, it was only 0.63. (Recall is difficult to differentiate for the two cases, since we do not know how many mentions in the gold standard are ambiguous.)

Other techniques applied: To further enhance system performance, especially in regard to false positive identification, we assembled a stop word list consisting of common English words, protein family terms, non-protein molecules, and experimental words, all of which are common distractor strings. The common English word stop list included 5,000 words derived by word frequency in the Brown corpus [13]. The protein family terms were derived from an in-house manual annotation project which annotated protein families. A list of small molecules, e.g. Ca2+, was also added.

Table 9: Results of gene normalization with different stop word lists.

Steps	Precision	Recall	F-measure
do not use stop list	0.764	0.739	0.752
use common English words stop list	0.776	0.738	0.757
use non-protein stop list	0.768	0.736	0.752
use custom stop list	0.811	0.730	0.769
use all three stop lists	0.827	0.727	0.774

Results on the test data: We submitted three separate runs for the GN task. Run 1 favored precision: it used all four stop lists and removed from the dictionary any terms that could be mapped to two or more identifiers. Run 2 aimed to optimize F-measure: it did not use the “protein family stop list,” and removed terms associated with three or more database identifiers. Run 3 aimed to

Table 10: Evaluation results by our system on the development set are shown in the first three rows. Row 4 shows the estimated recall ceiling for lexical matching reported by [22] for the same data set.

Run	True Positives	False Positives	False Negatives	Precision	Recall	F-measure
1	458	94	182	0.830	0.716	0.769
2	465	97	175	0.827	0.727	0.774
3	467	103	173	0.819	0.730	0.772
4	530	7941	110	0.063	0.828	0.117

Table 11: Results on the GN test data.

Run	True Positives	False Positives	False Negatives	Precision	Recall	F-measure	Quartile
1	576	109	209	0.841	0.734	0.784	1
2	583	120	202	0.829	0.743	0.784	1
3	587	129	198	0.820	0.748	0.782	1

optimize recall: it used fewer stop lists, and removed terms from the dictionary only when they could be mapped to five or more database identifiers. Table 10 shows that the results do not vary widely from the development set. We were able to improve on the estimated recall ceiling for simple matching to a lexicon as reported in [22].

Table 11 shows results on the test data. F-measure for all three runs is in the top quartile and is comparable to the highest F-measure (0.79) for the GN task in mouse (the most comparable of the three species in BioCreative 2004).

We believe the most innovative components of our system are (1) the approach for handling complex coordination properly, and (2) the rules for disambiguating among multiple gene matches for a particular string. This system is also unusual for a rule-based dictionary method in that it is (nearly) species-independent. We also note that the elimination of common distractor strings was particularly important in the performance of our system.

4 Protein Interaction Article Subtask (IAS)

We submitted three runs for the IAS subtask. These were generated by training machine-learning-based classifiers on linguistic and semantic features extracted from the training data. The most distinctive aspects of our approach to this task were 1) *Use of semantic features* and 2) *An attempt to balance the training set*. We noted a large discrepancy between our results on the training data and our results on the test data that we suspect reflects conceptual drift in the document collection. We discuss this at length at the end of this section.

We utilized the WEKA toolkit [33] to construct the ML-based classifiers. The features employed were n-grams (with n ranging from one to five) of stemmed words and matches to OpenDMAP patterns indicative of protein-protein interaction mentions (See Section 5). Table 12 summarizes the characteristics of the three classifiers that we built.

Balancing positives and negatives in the training data: For our third submission, we balanced the number of positive and negative training abstracts. (There were 3536 positive abstracts, compared with 1959 negative abstracts, in the training set.) We built an additional set of negative abstracts with characteristics similar to the positive abstracts by compiling a collection of verbs that are often used to describe genetic interactions (e.g. *enhance*, *express*, and *transactivate* and then querying MEDLINE with those terms. We narrowed that set down further by applying our Run-1 classifier to it, thus identifying abstracts which did not discuss protein-protein interaction, and added those articles to create an expanded training set with a 1:1 ratio of positive to negative abstracts. We trained a new classifier on this expanded training set and applied it to the test data to generate this submission.

Results: Our three classifiers achieved F-measures roughly equivalent to one another, and above, but

Table 12: The three classifiers used for the IAS subtask. *IG threshold* is the information gain feature selection threshold.

Name	Classifier		IG threshold
Run 1	SVM	RBF kernel, complexity factor 100, gamma 0.001	.0001
Run 2	Naïve Bayes	kernel estimation enabled	.001
Run 3	SVM with balanced +/-	RBF kernel, complexity factor 100, gamma 0.001	.0001

Table 13: IAS performance compared to the mean and median.

Run	Precision	Recall	F-measure	Accuracy	AUC
Run 1	0.699	0.853	0.768	0.743	0.754
Run 2	0.609	0.941	0.739	0.688	0.562
Run 3	0.706	0.813	0.756	0.737	0.752
Overall mean (standard deviation)	0.664 (0.081)	0.764 (0.193)	0.687 (0.104)	0.671 (0.064)	0.735 (0.074)
Median	0.677	0.851	0.722	0.668	0.752

within one standard deviation of, the overall mean. As in our cross-validation experiments on the training data, our first run achieved the best F-measure, but the difference in F-measures between the three are relatively small. The SVM classifiers (Runs 1 and 3) appear to achieve a higher precision and lower recall than the NB classifier, a characteristic that we have noticed in other document classification work where we compared these classification algorithms [2].

Discussion: We note that our IAS classifiers achieved much higher performance in cross validation on training data than on the test data. For example, our Run-1 classifier achieved a Precision of 0.951, a Recall of 0.945, and an F-measure of 0.948 in 10-fold cross validation of training data; the F-measure achieved in cross-validation was approximately 20% higher than that achieved on the test data. Cross-validation experiments are designed to minimize the effects of over-fitting, and our past experiences suggest that it is typically more successful than was indicated by this experiment. This suggests that a difference exists between the data compiled for the training set and the test set.

We analyzed the corpora and found that the publication years of the articles in the different sets showed that all of the positive training articles were published in either 2005 or 2006, while the negative training articles came from a wider distribution of publication years, centered around 2001. Only about 10% of the negative training articles were published in 2005 or 2006, so it is possible that our classifiers discriminated partially based on the publication years (possibly represented in the feature sets, for example, by a bias in the types of experimental procedures mentioned). Our Run-1 system expressed a bias toward positive classification on the test set (458 positive classifications and 292 negative classifications), where 91% of the articles were published in 2006.

A. Cohen et al. (2004) noted a similar phenomenon in the TREC 2004 Genomics track data. Our short analysis supports the findings of their more extensive study. The difference that they noted was substantially smaller than the one that we report here—about 12%, versus the approximately 20% that we report—suggesting that the training/test data for BioCreative might represent a good data set for working on this problem.

While this apparent publication year bias appears to be an issue with the construction of the training and test corpora, it represents a real-world problem that needs to be dealt with if we are to develop truly useful machine-learning-based document classification systems. Since ideally we would train on currently available data, and apply our systems to literature as it is published, we would require that systems not be affected by this type of bias. A concept-based approach where terms are recognized and mapped to an ontology, as opposed to a purely linguistic-based approach, as we

employed for these classifiers, might help avoid over-fitting of classifiers to development data sets. For example, we found that terms describing experimental approaches for detecting protein-protein interactions (e.g., *yeast two hybrid*, *two dimensional gel electrophoresis*, *coimmunoprecipitation*, and *MALDI-TOF*) were among the most important features in discriminating positive from negative articles. The useful information in these features is not the mention of a specific experimental method, but the fact that a technique for recognizing protein-protein interactions was mentioned. (As one reviewer suggested, this points out the value of having a knowledge model that reflects the curation criteria for the reference databases, since they only contain experimentally confirmed interactions.) This is consistent with the hypothesis (advanced by us and others elsewhere, e.g. [2, 3, 6]) that a better approach when training classifiers is to attempt to map words to their underlying concepts. Using this approach, we hypothesize that future systems would be more scalable and robust.

5 Protein Interaction Pairs Subtask (IPS)

For the IPS subtask we used the OpenDMAP (Open source Direct Memory Access Parsing) semantic parser. As is typical for semantic parsers using manually-constructed grammars, our system is geared towards optimizing precision. The procedure begins with preprocessing the HTML, then moves to species recognition, entity tagging, and part of speech tagging, followed by extraction of protein-protein interactions. Our approach to detecting interacting protein pairs relies heavily on the systems generated for the GM and GN tasks.

5.1 Preprocessing

HTML Parsing: The HTML parser developed to process the raw HTML documents was an extension of a similar parser developed for the TREC Genomics 2006 task [3]. Embedded HTML tags are removed, and images representing Greek characters are converted to ASCII strings. The title, abstract, paragraphs, sentences, section headings, and sub-section headings were extracted for each document. Document sections are inferred based on the section heading text. Sentence boundaries are detected using the LingPipe sentence chunker [4]. Sentences are mapped back to the original HTML using a dynamic programming approach.

Gene Mention Tagging: We used a variant of the system developed for the GM task to tag genes. For the IPS task, the outputs of ABNER [28] (both models) and LingPipe [4] (BioCreative04 model) were combined using the *overlapping filter* (See Section 2).

Part of Speech Tagging was done with the GENIA POS Tagger [31].

Species Classification was done with a modified dictionary search. The dictionary was constructed from the intersection of words from the NCBI *names.dmp* file (a list of all known scientific names and synonyms for organisms) and the set of NCBI taxonomy identifiers present in the IPS training set. These words were then combined into a single regular expression pattern for each species. Some filtering of false positive species mentions was achieved by evaluating bigrams present in the flanking regions of each species mention. Each species detected in an article was given a score based on the number of times it appeared and results of its flanking region evaluation, and a ranked list of species for a given article was returned. The BioCreative 2006 PPI training set was used to create a list of bigrams present in the flanking region (± 50 character positions) of each species pattern match. These “indicator bigrams” were each assigned a log-odds score corresponding with the formula:

$$p(\text{bigram occurs in the flanking region of a true positive dictionary match}) / p(\text{bigram occurs in the flanking region of a false positive dictionary match})$$

The species patterns found in each test article were given scores according to the sum of all log-odds scores for indicator bigrams found in the flanking region. The total score for a given species classification for a single article was calculated by summing all individual pattern match scores. Once scored, the species for a given document are returned in rank order. We experimented with the optimal number of species results to return and found the best results when the maximum number of species returned from the ranked list was two.

5.2 Gene Mention Normalization

Gene Lexicon Construction: Dictionaries were constructed for the IPS task for each species that was observed in the training data by extracting information from the `uniprot_light_table_updated.txt` file supplied by the BioCreative organizers.

Gene Mention Normalization: Each gene mention was normalized using the procedure described above for the GN task, using the dictionary for the apparent species. We experimented with the optimal number of normalized identifiers to return and found the best results when we limited the output to one normalized entry per gene mention in text.

5.3 OpenDMAP and Conceptual Patterns

OpenDMAP patterns are written in a context-free syntax. Non-terminal elements are defined in a Protégé ontology. A simple example of an OpenDMAP pattern for the IPS task looks like:

{*interaction*} → [*interactor1*] interacts with [*interactor2*];

... where elements in {braces} represent classes in the ontology, elements in [brackets] correspond to slots of the class on the left-hand side of the pattern, and bare strings are terminals. The slots are constrained in the ontology to have specific features; for the IPS task, the slot elements [*interactor1*] and [*interactor2*] are constrained to be proteins. The output is sentences in which OpenDMAP found text matching a protein interaction pattern, as well as the entities involved in the interaction.

OpenDMAP patterns allow for recursion and for the free mixing of terminals and non-terminals. For instance, the following patterns:

{*interaction*} → {*interact-noun*} {*preposition*} {*determiner*}? {*protein-list*} with {*protein-list*};
{*protein-list*} → [*interactor1*] and [*interactor2*];

... match the bolded text in *The present report examines protein-protein **interaction of NMT1 and NMT2 with m-calpain and caspase-3** in human colorectal adenocarcinoma tissues and HC-CLs* (PMID 16530191) and returns the four interacting pairs NMT1/m-calpain, NMT1/caspase-3, NMT2/m-calpain, and NMT2/caspase-3.

We used a variety of discovery procedures to build the grammar, including scheduled elicitation sessions with “native speakers” (scientists with expertise in biology) and examination of corpora for frequently-occurring ngrams and frequently-occurring strings between protein mentions [25]. We used the BioCreative 2006 IPS, ISS and IAS training data, the PICorpus⁴ [1, 15], material generated by Jörg Hakenberg [23]⁵ and Anna Veuthey, and the Prodisen corpus⁶. We tuned the system using a 40,000-sentence subset of the IPS training data. The final grammar consisted of 67 rules. It handles verbs, nominalizations, and various forms of conjunction, but not negation.

5.4 Results

There was a marked difference between our performance on the training data and on the test data. Our results on the training data were $P = 0.364$, $R = 0.044$, and $F = 0.078$, returning 385 pairs. However, we achieved recall as high as 0.31 on the test data (seven times higher than on the training data), and recall higher than the median on 2 of 5 measures (see Tables 14 and 15). Our F-measure was above the median more often than it was below it.

6 Protein Interaction Sentences Subtask (ISS)

We modelled the ISS subtask as a summarization task, using an approach similar to the Edmundsonian paradigm: we created a scoring scheme to rate sentences as either containing an interaction mention, or not. This approach has been used for selecting candidate GeneRIFs from Medline abstracts [18].

⁴ Available at <http://bionlp.sourceforge.net/>

⁵ Available at <http://www2.informatik.hu-berlin.de/~hakenber/>

⁶ Available at <http://www.pdg.cnb.uam.es/martink/PRODISEN/>

Table 14: Comparison of interaction pairs results

	calculated by interaction			calculated by article		
	P	R	F	P	R	F
Run 1	0.38	0.06	0.11	0.39	0.31	0.29
Median	0.06	0.11	0.07	0.08	0.20	0.08

Table 15: Comparison of normalization

	calculated by interactor			calculated by article			calculated by article with interactions		
	P	R	F	P	R	F	P	R	F
Run 1	0.57	0.12	0.19	0.15	0.13	0.13	0.56	0.46	0.48
Median	0.18	0.25	0.19	0.16	0.28	0.17	0.21	0.39	0.24

Task Data: Table 16 shows the size of the development and blind test data sets. The training set includes a total of 29 full-text articles with 53 gold-standard sentences selected by IntAct⁷ and MINT⁸ database curators. On average, in the development set, there are approximately two sentences per article, which is markedly smaller than the average number of sentences (5.5) per article in the test set. We did not make use of the number of interaction sentences per article in the development set; systems that did would be likely to undergenerate.

Sentence Selection: Each candidate interaction sentence is scored based on criteria which differ depending on the location of the sentence in the document⁹ (Table 17). In order to be scored, the sentence first must meet certain eligibility requirements.

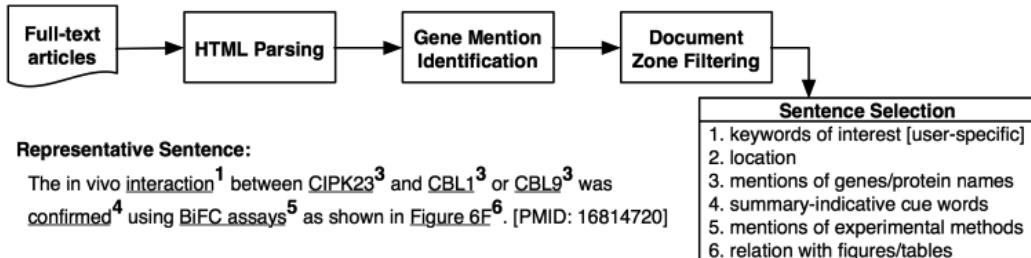


Figure 1: General system design for finding protein interaction sentences from full-text articles. A sample gold standard sentence is shown on the bottom left with key scoring components underlined and numbered according to the corresponding sentence selection category.

Scoreable Features: Our system scores each sentence in a full-text article with respect to these features (Figure 1):

1. Frequent words: The frequent words in the gold standard are all related to protein-protein interaction. For instance, the word *interact* and the phrase *interaction of* are the most frequent unigram and bigram, respectively.
2. Location: We found most gold-standard passages in the *Results* section, and few in the *Title*, *Abstract* or *Introduction* sections. Some sections never yielded a sentence.
3. Mentions of gene/protein names: Since the sentences make assertions about protein-protein interaction, protein mentions are necessary in these sentences.

⁷<http://www.ebi.ac.uk/intact>

⁸<http://mint.bio.unroma2.it/mint/>

⁹Section-specific usefulness and error rates have been noted in other BioNLP application areas, e.g. [30].

Table 16: BioCreative II protein interaction sentences (ISS) task: development and test data sets.

Data set	articles	sentences	interaction sentences/article
Development (July release)	9	24	2.67
Development (Sept. release)	20	39	1.95
Development (Sum)	29	53	1.83
Blind test	358	1,978	5.53

Table 17: **Scoring requirements** P: Has positive cue words, N: Does not have negative cue words, G: has >0 gene mentions, X: has experimental methods, I: has interaction key word; * If a sentence includes a reference to a figure or table, the score for the caption is added to the score for the sentence.

Location	Requires					Scored on				
	P	N	G	X	I	P	N	G	X	I
Abstract	✓	✓							✓	✓
Figure/Table Caption	✓	✓	✓	✓					✓	✓
Section/Subsection Heading	✓			✓					✓	✓
Other*	✓	✓		✓	✓				✓	✓

4. Summary-indicative cue words: Words (e.g. *confirm*) or phrases (e.g. *data establish*) that indicate a sentence is likely to be a good interaction sentence.
5. Mentions of experimental methods: Protein-protein interaction detection methods (e.g. *two hybrid array*) are frequently mentioned in the gold-standard passages.
6. Figure/table mention: Many gold-standard passages refer to a table or figure.

Preprocessing: The methods used for HTML parsing and gene name tagging were the same as used for the IPS task (See Section 5). In an attempt to remove false positives prior to processing, we implemented a document zoning filter which excluded sentences associated with certain document sections. The excluded document sections were chosen from manual inspection of some of the training data. The sections include: *Materials and Methods*, *Acknowledgments*, *Discussion*, *Reference*, *Table of Contents*, *Disclosures*, and *Glossary*.

Results: We submitted two runs for the ISS task. The runs differed only in the passage length returned for each “interaction sentence.” For our Run #1, the returned passage was limited to a single sentence. This restriction was loosened for Run #2, permitting multiple consecutive high-scoring sentences to be returned.

Our results show that loosening the passage length restriction permitted the extraction of 39.2% more passages that had been pre-selected by the human curators when compared to our single-sentence run (Table 18). This suggests that informative sentences regarding protein interactions in full text are likely to be found in close proximity. This contrasts with the case of abstracts, in which such sentences tend to be found at opposite ends of the text [18]. Note that we made no attempt to rank our outputs.

7 Discussion

Preliminary results suggest that the BioCreative 2006 PPI materials might be a fruitful data set for investigating the issues of conceptual drift raised by [8].

A major goal of our work on this shared task was to extend the OpenDMAP semantic parser. We did so, incorporating a number of third-party linguistic and semantic analysis tools without sur-

Table 18: ISS results. Runs 1 and 2 are our submissions. *Passages*, the total number of passages evaluated; *TP*, the number of evaluated passages that were pre-selected by human curators; *Unique*, the number of unique passages evaluated. *U_TP*, the number of unique passages that were pre-selected; *MRR*, mean reciprocal rank of the correct passages.

Run	Passages	TP	Unique	U_TP	TP/Passages	U_TP/Unique	MRR
Run #1	372	51	361	51	13.71	14.13	1.0
Run #2	372	71	361	70	19.09	19.39	1.0

rendering an essential characteristic of the DMAP paradigm: complete integration of semantic and linguistic knowledge, without segregating lexical and domain knowledge into separate components.

We used UIMA [11, 20] as a framework for integrating the various software components used throughout our BioCreative 2006 submissions. For each major component, a UIMA wrapper was created so that it could be plugged into the system. For the GM task, a UIMA wrapper was created for each gene tagger. A component for reading in the document collection was also created, as was a component for outputting the results into the format required by the *alt_eval.pl* script. The output component was also crucial for converting the annotation spans created by the taggers into the somewhat idiosyncratic output format required by the competition organizers. Using the UIMA framework enabled our system to quickly convert between the two different filters, *consensus* and *overlapping*, by simply swapping out the components, and to evaluate their effects quite quickly.

By using a standardized framework, we were not only able to distribute the tasks of development with the assurance that the pieces would work in concert once combined, but we were also able to design our systems in such a way that as they became successively more complicated, evaluation remained quick, easy, and modular. Not only was it possible to incorporate infrastructure constructed expressly for the BioCreative tasks, but it was just as easy able to utilize external tools developed prior to the BioCreative tasks and/or by third-parties. This allowed us to benefit from LingPipe, Schwartz and Hearst’s abbreviation-defining algorithm, ABNER, KeX, ABGene, and the GENIA POS tagger (op cit). Utilizing this framework provided not only a robust development architecture and production-ready execution environment, but also a tremendous time savings.

8 Acknowledgments

We thank the two anonymous BioCreative reviewers for their insightful feedback. This work was supported by NIH grant R01-LM008111 to Larry Hunter.

References

- [1] Christian Blaschke, Miguel A. Andrade, Christos Ouzounis, and Alfonso Valencia. Automatic extraction of biological information from scientific text: protein-protein interactions. In *Intelligent Systems for Molecular Biology 1999*, pages 60–67, 1999.
- [2] Gregory J. Caporaso, William A. Baumgartner, Jr., Bretonnel K. Cohen, Helen L. Johnson, Jesse Paquette, and Lawrence Hunter. Concept recognition and the TREC Genomics tasks. In *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*, 2005.
- [3] JG Caporaso, WA Baumgartner, Jr., H Kim, Z Lu, HL Johnson, O Medvedeva, A Lindemann, LM Fox, EK White, KB Cohen, and L Hunter. Concept recognition, information retrieval, and machine learning in genomics question answering. In *Proceedings of The Fifteenth Text REtrieval Conference (TREC 2006)*, 2006.
- [4] Bob Carpenter. Phrasal queries with LingPipe and Lucene: ad hoc genomics text retrieval. In *Proceedings of the 13th annual Text Retrieval Conference*, 2004.

- [5] Lifeng Chen, Hongfang Liu, and Carol Friedman. Gene name ambiguity of eukaryotic nomenclatures. *Bioinformatics*, 21(2):248–56, 2005.
- [6] A. M. Cohen. Unsupervised gene/protein entity normalization using automatically extracted dictionaries. *Proceedings of the BioLINK2005 Workshop Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, pages 17–24, 2005.
- [7] Aaron M. Cohen. Unsupervised gene/protein named entity normalization using automatically extracted dictionaries. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, pages 17–24, 2005.
- [8] Aaron M. Cohen, Ravi Teja Bhupatiraju, and William R. Hersh. Feature generation, feature selection, classifiers, and conceptual drift for biomedical document triage. In *Proceedings of the 13th Text Retrieval Conference*, 2004.
- [9] KB Cohen, AE Dolbey, GK Acquaah-Mensah, and L Hunter. Contrast and variability in gene names. In *Proceedings of ACL Workshop on Natural Language Processing in the Biomedical Domain*, pages 14–20, 2002.
- [10] H Fang, K Murphy, Y Jin, J S Kim, and P S White. Human gene name normalization using text matching with automatically extracted synonym dictionaries. In *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology*, pages 41–48, 2006.
- [11] David Ferrucci and Adam Lally. UIMA: an architectural approach to unstructured information processing in the corporate. *Nat. Lang. Eng.*, 10(304):327–348, 2004.
- [12] W. Fitzgerald. *Building Embedded Conceptual Parsers*. PhD thesis, Northwestern University, 1994.
- [13] WN Francis and H Kucera. *Brown Corpus Manual*. Brown University, 1964.
- [14] K Fukuda, A Tamura, T Tsunoda, and T Takagi. Toward information extraction: identifying protein names from biological papers. *Pac Symp Biocomput*, pages 707–18, 1998.
- [15] Helen L. Johnson, William A. Baumgartner, Jr., Martin Krallinger, K. Bretonnel Cohen, and Lawrence Hunter. Refactoring corpora. In *Proceedings of the BioNLP workshop on linking natural language processing and biology at HLT-NAACL 06*, pages 116–117, 2006.
- [16] Shuhei Kinoshita, K. Bretonnel Cohen, Philip V. Ogren, and Lawrence Hunter. BioCreAtIvE task1A: entity identification with a stochastic tagger. *BMC Bioinformatics*, 6 Suppl 1(1471-2105 (Electronic)):S4, 2005.
- [17] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 SIGDOC Conference*, pages 24–26, 1987.
- [18] Z Lu, KB Cohen, and L Hunter. Finding GeneRIFs via Gene Ontology annotations. *Pac Symp Biocomput*, pages 52–63, 2006.
- [19] Zhiyong Lu. *Text mining on GeneRIFs*. PhD thesis, University of Colorado School of Medicine, 2007.
- [20] R. Mack, S. Mukherjea, A. Soffer, N. Uramoto, E. Brown, A. Coden, J. Cooper, A. Inokuchi, B. Iyer, Y. Mass, H. Matsuzawa, and L.V. Subramaniam. Text analytics for life science using the Unstructured Information Management Architecture. *IBM Systems Journal*, 43:490–515, 2004.

- [21] C. E. Martin. *Direct Memory Access Parsing*. PhD thesis, Yale University, 1991.
 - [22] AA Morgan, B Wellner, JB Colombe, R Arens, ME Colosimo, and L Hirschman. Evaluating the automatic mapping of human gene and protein mentions to unique identifiers. *Pac Symp Biocomput*, pages 281–291, 2007.
 - [23] Conrad Plake, Joerg Hakenberg, and Ulf Leser. Optimizing syntax patterns for discovering protein-protein interactions. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 195–201, New York, NY, USA, 2005. ACM Press.
 - [24] MF Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
 - [25] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proc. 40th annual meeting of the Association for Computational Linguistics*, pages 41–47, 2002.
 - [26] CK Riesbeck. From conceptual analyzer to Direct Memory Access Parsing: an overview. In NE Sharkey, editor, *Advances in Cognitive Sciences*. Ellis Horwood Limited, 1986.
 - [27] Ariel S Schwartz and Marti A Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. *Pac Symp Biocomput*, pages 451–62, 2003.
 - [28] Burr Settles. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–2, 2005.
 - [29] Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–32, 2002.
 - [30] Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in full text articles. In *Natural language processing in the biomedical domain*, pages 9–13. Association for Computational Linguistics, 2002.
 - [31] Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun’ichi Tsuji. Developing a robust part-of-speech tagger for biomedical text. In *Advances in informatics—10th Panhellenic Conference on Informatics*, pages 382–392, 2005.
 - [32] O Tuason, L Chen, H Liu, J A Blake, and C Friedman. Biological nomenclatures: a source of lexical knowledge and ambiguity. *Pac Symp Biocomput*, pages 238–49, 2004.
 - [33] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, June 2005.
 - [34] Alexander Yeh, Alexander Morgan, Marc Colosimo, and Lynette Hirschman. BioCreAtIVe task 1A: gene mention finding evaluation. *BMC Bioinformatics*, 6 Suppl 1(1471-2105 (Electronic)):S2, 2005.



Adapting a Relation Extraction Pipeline for the BioCreAtIvE II Tasks

Claire Grover¹

grover@inf.ed.ac.uk

Barry Haddow¹

bhaddow@inf.ed.ac.uk

Ewan Klein¹

ewan@inf.ed.ac.uk

Michael Matthews¹

m.matthews@ed.ac.uk

Leif Arda Neilsen¹

lnielsen@inf.ed.ac.uk

Richard Tobin¹

richard@inf.ed.ac.uk

Xinglong Wang¹

xwang@inf.ed.ac.uk

¹ School of Informatics, University of Edinburgh, 2 Buccleuch Place, Edinburgh, EH8 9LW, Scotland

Abstract

The Second BioCreAtIvE Challenge provided an ideal opportunity to evaluate biomedical NLP techniques. Prior to the Challenge, an information extraction pipeline was developed to extract entities and relations relevant to the biomedical domain, and to normalise the entities to appropriate ontologies. With minimal effort, the pipeline was adapted to work with the BioCreAtIvE data and achieved results that appear competitive with existing state-of-the-art systems.

Keywords: biomedical NLP, relation extraction, named entity recognition, term identification

1 Introduction

The team 6 (T6) submissions for BioCreAtIvE II were based on research carried out as part of the TXM programme, a three year project aimed at producing NLP tools to assist in the curation of biomedical research papers. The principal product of this project is an information extraction pipeline, designed to extract entities and relations relevant to the biomedical domain, and to normalise the entities to appropriate ontologies. The submissions for the BioCreAtIvE II protein-protein interaction subtasks (IPS, ISS and IAS) used the output of the pipeline directly, whilst the submissions for the GM and GN tasks used techniques developed during the implementation of the pipeline. It was found that the TXM information extraction pipeline could be used without modification on the BioCreAtIvE II data, and appeared to maintain a similar level of performance as on the TXM test data.

For the phase 1 release of the TXM pipeline, the focus was on the recognition of protein mentions, protein-protein interactions (PPIs) and the normalisation of the proteins to a RefSeq-derived wordlist. In order to render the pipeline easily adaptable to other domains, machine learning approaches were favoured, and consequently a large quantity of annotated data was produced to train the system and to test its performance.

In Section 2, the information extraction pipeline is described, as well as the methods used in each of the five T6 submissions. Section 2 provides a brief analysis of the results for each submission, with an attempt to identify the major sources of error.

2 Methods

2.1 The TXM Information Extraction Pipeline

The TXM pipeline consists of a series of natural language processing tools, integrated within the LT-XML2 architecture.¹ In order to train and test the pipeline, we used a corpus of 151 full-texts and 749 abstracts which had been selected from PubMed and PubMedCentral as containing experimentally determined protein-protein interactions. The corpus was annotated by trained biologists for proteins and related entities, protein normalisations (to an in-house wordlist derived from RefSeq) and protein-protein interactions. Around 80% of the documents were used for training and optimising the pipeline, while the other 20% were held back for testing.

The major components of the pipeline are as follows:

Preprocessing The preprocessing component comprises tokenisation, sentence boundary detection, lemmatisation, part-of-speech tagging, species word identification, abbreviation detection and chunking. The part-of-speech tagging uses the Curran and Clark maximum entropy Markov model tagger [2] trained on MedPost data [16], whilst the other preprocessing stages are all rule-based. We implemented tokenisation, sentence boundary detection, species word identification and chunking with the LT-XML2 tools. For abbreviation extraction, we used the Schwartz and Hearst abbreviation extractor [14] and for lemmatisation we employed `morphe` [12].

Named Entity Recognition In the pipeline, named entity recognition (NER) of proteins is performed using the Curran and Clark classifier [2], augmented with extra features tailored to the biomedical domain.

Term Normalisation The term normalisation task in the pipeline involves choosing the correct identifier for each protein mention in the text, where the identifiers are drawn from a lexicon based on RefSeq. A set of candidate identifiers is generated using hand-written fuzzy matching rules, from which a single best identifier is chosen using a machine-learning based species tagger, and a set of heuristics to break ties. The term normalisation component of the pipeline was not used directly in the BioCreAtIvE II tasks since they employ different protein lexicons.

Relation Extraction To find the PPI mentions in the text, we built a maximum entropy relation extractor trained using shallow linguistic features [13]. The features include context words, parts-of-speech, chunk information, interaction words and interaction patterns culled from the literature. The relation extractor examines each pair of proteins mentioned in the text, and occurring less than a configurable number of sentences apart, and classifies them as being in an interaction or not. Whilst the relation extractor can theoretically recognise both inter-sentential and intra-sentential relations, since both types of candidate relations are considered, in practice very few inter-sentential relations are correctly recognised. Only around 5% of annotated relations are inter-sentential, and it is likely that using exactly the same techniques as on the intra-sentential relations is not optimal, especially since many of the inter-sententials use coreferences. The detection of inter-sentential relations is the subject of ongoing research.

In the remainder of this section, we will describe how this pipeline was deployed for carrying out the T6 submissions.

¹<http://www.ltg.ed.ac.uk/software/xml/>

2.2 Gene Mention Task

To address the Gene Mention (GM) task, T6 employed two different machine learning methods using similar feature sets. Runs 1 and 3 used conditional random fields (CRF) [8], whilst run 2 used a bidirectional maximum entropy Markov model (BMEMM) [19].

Both CRF and BMEMM are methods for labelling sequences of words which model conditional probabilities so that a wide variety of possibly inter-dependent features can be used. The named entity recognition problem is represented as a sequential word tagging problem using the BIO encoding, as in CoNLL 2003 [18]. In BMEMM, a log-linear feature-based model represents the conditional probability of each tag, given the word and the preceding and succeeding tags. In CRF, by contrast, the conditional probability of the whole sequence of tags (in one sentence), given the words, is represented using a log-linear model. Both methods have been shown to give state-of-the-art performance in sequential labelling tasks such as chunking, part-of-speech-tagging and named entity recognition [10, 11, 15, 19]. The CRF tagger was implemented with CRF++² and the BMEMM tagger was based on Zhang Le's MaxEnt Toolkit.³

GM Preprocessing Before training or tagging the documents with the machine learner, we passed them through the preprocessing stages of the TXM pipeline (see Section 2.1).

GM Features For the machine learners, we extracted the following features for each word:

word The word itself is added as a feature, plus the four preceding words and four succeeding words, with their positions marked.

headword The headwords of noun and verb phrases are determined by the chunker, and, for all words contained in noun phrases, the head noun is added as a feature.

affix The affix feature includes all character n -grams with lengths between two and four (inclusive), and either starting at the first character, or ending at the last character of the word.

gazetteer The gazetteer features is calculated using an in-house list of protein synonyms derived from RefSeq. To add the gazetteer features to each word in a given sentence, the gazetteer is first used to generate a set of matched terms for the sentence, where each word is only allowed to be in one matched term and earlier starting, longer terms take precedence. The unigram gazetteer feature for each word has value either B, I or O, depending on whether the word is at the beginning, inside or outside of a gazetteer matched term. The bigram gazetteer feature is also added, and this is the concatenation of the previous and current word's gazetteer feature.

character For each of the regular expressions listed in Table 1, the character feature indicates whether or not the word matches the regular expression. These regular expressions were derived from lists published in previous work on biomedical and newswire NER [1, 2]. The length of the word is also included as a character feature.

postag This feature includes current word's part-of-speech tag and the POS tags for the two preceding and succeeding words. Also added are the bigram of the current and previous word's POS tag, and the trigram of the current and previous two words' POS tags.

wordshape The word shape feature consists of the word type feature of [2], a variant of this feature which only collapses runs of greater than two characters in a word, and bigrams of the word type feature.

abbreviation The abbreviation feature is applied to all abbreviations whose antecedent is found in the gazetteer.

²<http://chasen.org/~taku/software/CRF++/>

³http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

Description	Regexp
Capitals, lower case, hyphen then digit	[A-Z]+[a-z]*-[0-9]
Capitals followed by digit	[A-Z]{2,}[0-9]+
Single capital	[A-Z]
Single Greek character	\p{InGreek}
Letters followed by digits	[A-Za-z]+[0-9]+
Lower case, hyphen then capitals	[a-z]+-[A-Z]+
Single digit	[0-9]
Two digits	[0-9][0-9]
Four digits	[0-9][0-9][0-9][0-9]
Two capitals	[A-Z][A-Z]
Three capitals	[A-Z][A-Z][A-Z]
Four capitals	[A-Z]{4}
Five or more capitals	[A-Z]{5,}
Digit then hyphen	[0-9]+-
All lower case	[a-z]+
All digits	[0-9]+
Nucleotide	[AGCT]{3,}
Capital, lower case then digit	[A-Z][a-z]{2,}[0-9]
Lower case, capitals then any	[a-z][A-Z][A-Z].*
Greek letter name	Match any Greek letter name
Roman digit	[IVXLCD]+
Capital, lower, capital and any	[A-Z][a-z][A-Z].*
Contains digit	.*[0-9].*
Contains capital	.*[A-Z].*
Contains hyphen	.*-.*
Contains period	.*\..*
Contains punctuation	.*\p{Punct}.*
All digits	[0-9]+
All capitals	[A-Z]+
Is a personal title	(Mr Mrs Miss Dr Ms)
Looks like an acronym	([A-Za-z]\.)+

Table 1: The (Java) regular expressions used for the character feature in the GM task.

2.3 Gene Normalisation Task

The Gene Normalisation (GN) system was developed with genericity in mind. In other words, it can be ported to normalise other biological entities (e.g., disease types, experimental methods, etc) relatively easily, without requiring extensive knowledge of the new domain. The approach that was adopted combined a string similarity measure with machine learning techniques for disambiguation.

For GN, our system first preprocesses the documents (see Section 2.1) and then uses the gene mention NER component (see Section 2.2) to mark up gene and gene product entities in the documents. A fuzzy matcher then searches the gene lexicon provided and calculates scores of string similarity between the mentions and the entries in the lexicon using a measure similar to JaroWinkler [5, 6, 20].

The Jaro string similarity [5, 6] measure is based on the number and order of characters that are common to two strings. Given strings $s = a_1 \dots a_k$ and $t = b_1 \dots b_l$, define a character a_i in s to be *shared with* t if there is a b_j in t such that $b_j = a_i$ with $i - H \leq j \leq i + H$, where $H = \frac{\min(|s|, |t|)}{2}$. Let $s' = a'_1 \dots a'_{k'}$ be the characters in s which are shared with t (in the same order as they appear in s) and let $t' = b'_1 \dots b'_{l'}$ be analogous. Now define a *transposition* for s', t' to be a position i such that

$a'_i \neq b'_j$. Let $T_{s',t'}$ be half the number of transpositions for s' and t' . The Jaro similarity metric for s and t is shown in Equation 1.

$$\text{Jaro}(s, t) = \frac{1}{3} \cdot \left(\frac{|s'|}{s} + \frac{|t'|}{t} + \frac{|s'| - T_{s',t'}}{|s'|} \right) \quad (1)$$

A variant of the Jaro measure due to Winkler [20] also uses the length P of the longest common prefix of s and t . It rewards strings which have a common prefix. Letting $P' = \max(P, 4)$, it is defined as shown in Equation 2:

$$\text{JaroWinkler}(s, t) = \text{Jaro}(s, t) + \frac{P'}{10} \cdot (1 - \text{Jaro}(s, t)) \quad (2)$$

For the GN task, a variant of the JaroWinkler measure was employed, as shown in Equation 3, which uses different weighting parameters and takes into account the suffixes of the strings.

$$\text{JaroWinkler}'(s, t) = \text{Jaro}(s, t) + \min(0.99, \frac{P'}{10} + \theta) \cdot (1 - \text{Jaro}(s, t)) \quad (3)$$

Here, $\theta = (\# \text{CommonSuffix} - \# \text{DifferentSuffix})/\text{lengthOfString}$. The idea is not only to look at the common prefixes but also commonality and difference in string suffixes. A set of equivalent suffix pairs was defined, for example, the Arabic number 1 is defined as equivalent to the Roman number *I*. The number of common suffixes and the number of different suffixes (e.g., 1 and 2 or 1 and *II* would count as different suffixes) is counted, and strings with common suffixes are rewarded whilst those with different ones are penalised. The value is finally normalised by the length of the string.

At the end of the fuzzy-matching stage, each mention recognised by NER is associated with the single highest scoring match from the gene lexicon, where the score indicates the string similarity. Note that each match is associated with one or more identifiers (i.e., in cases where ambiguity occurs) from the gene lexicon.

The GN system collects all the gene identifiers, where every gene identifier is paired up with a set of features. These identifier-featureset pairs are used as training data to learn a model that predicts the most probable identifier out of a pool of candidates returned by the fuzzy matcher. Feature selection was manually carried out and simple features include the contextual text properties surrounding the mentions such as adjacent words, their part-of-speech tags, etc., and complex features such as the distance scores between the mentions in text and the matches returned by the fuzzy matcher. It turned out that the complex features are particularly helpful in terms of increasing the F_1 score.

In more detail, all the identifiers in a document found by the fuzzy matcher were collected, then the ones that are correct according to the answer file were used as positive examples and the others were used as negative ones. Each identifier was associated with a set of features as follows:

fuzzy-confidence Confidence scores⁴ from the fuzzy matcher.

synonym-similarity The averaged confidence score of the similarity between all synonyms linked to the gene identifier and the match.

context-similarity The similarity between descriptions (ie., synonyms) associated with a gene identifier and all gene entities in the current document recognised by the NER. The similarity is calculated by two measures: Dice coefficient⁵ and $tf * idf$.⁶

ner-confidence Confidence score generated by the NER tagger.

⁴Only those matches with confidence scores higher than 0.80 were considered.

⁵Dice coefficient is defined as twice the number of common terms in the two sets of tokens to compare, divided by the total number of tokens in both sets, ie., $\text{Dice} = \frac{2 * \text{commonTerms}}{\# \text{of terms in set 1} + \# \text{of terms in set 2}}$.

⁶ $tf * idf$ is defined as the product of term frequency (tf) and inverse document frequency (idf). $tf_i = \frac{n_i}{\sum_k n_k}$, where n_i is the number of occurrences of the considered term and the denominator is the number of occurrences of all terms. $idf_i = \log \frac{|D|}{\#\{d: d \ni t_i\}}$, where $|D|$ is the total number of documents and the denominator is the number of documents where the term t_i appears.

context Local features, including contextual words (± 10),⁷ lemmas (± 4), POS tags (± 2), species words (± 10) and bigrams (± 5).

length Length of the gene mention and length of the match.

With the positive and negative examples extracted, determining the correct normalisations becomes a standard machine learning task. A classifier using SVM^{light} [7] was trained on the examples extracted from the BioCreAtIvE II GN training data.

The documents were processed similarly for the testing. In detail, a document was first run through the NER tagger where all the potential entities were marked up. The fuzzy matcher then searched the gene lexicon and produced a list of candidate gene identifiers, which were associated with the features extracted from the context of the document and classified using the SVM model trained in the training stage. Finally, the positive identifiers predicted by the model were output as the correct normalisations of the document.

2.4 Interaction Article Subtask

The Interaction Article Subtask (IA) was treated as a standard document classification problem where abstracts were classified as CURATABLE if they contained curatable protein interaction information and NOT-CURATABLE otherwise. Document classification techniques typically use a bag-of-words approach which ignores the word order in the document. This approach was extended by using a ‘bag-of-nlp’ approach where in addition to words, a variety of features derived from the output of a natural language processing (NLP) pipeline were added to the bag. The classification was performed with SVM^{light} [7] using the linear kernel with the default parameters. The documents were ordered based on the output from the SVM classifier.

IA Preprocessing Before the documents were passed to the machine learner for training or classification, they were first passed through the TXM pipeline (see Section 2.1). In addition, each of the named-entities and compound nouns in the document were marked as phrases.

IA Features The features extracted for each document are described below. Only features that occurred at least twice in the training data were used and each feature was given a binary weight. Each feature was converted to lowercase and words found in a custom stopword list were ignored. For each word and word stem, BACKOFF and BACKOFF-STEMMED versions were also calculated by converting all numbers to a single ‘#’ symbol and removing all punctuation.

word The word itself.

word-backoff The BACKOFF version of the word.

bigram The bigrams of the BACKOFF feature. The bigrams were not allowed to cross sentence boundaries.

chunk The concatenation of the BACKOFF-STEMMED version of each word in a chunk up to a maximum of seven words.

phrase The concatenation of the BACKOFF-STEMMED version of each word in a phrase (one-word phrases were included).

phrase-bigram The bigrams of the PHRASE feature. All proteins were converted to the token NER-PROTEIN. The bigrams were not allowed to cross sentence boundaries.

chunk-headword-bigram The bigrams of the BACKOFF-STEMMED version of each headword of successive chunks. Chunks containing negative phrases (e.g., does not interact) were indicated by preceding the bigram with NEG.

⁷The numbers in parentheses denote the size of the context window.

chunk-headword-trigram The trigrams of the BACKOFF-STEMMED version of each headword of successive chunks. All proteins were converted to the token NERPROTEIN. Chunks containing negative phrases were indicated by preceding the trigram with NEG.

protein Added if the document contained at least one protein.

two-proteins Added if the document contained at least two unique proteins.

no-proteins Added if the document did not contain any proteins.

title-proteins Added if the document contained two unique proteins in the title.

2.5 Interaction Pair Subtask

The T6 Interaction Pair (IP) Subtask system made use of the TXM information extraction pipeline to identify mentions of protein-protein interactions (PPIs), together with additional components to normalise proteins to UNIPROT and to identify the curatable interactions from amongst the interaction mentions.

Data Preparation Two methods of data preparation were used. In runs 1 and 3, the supplied pdftotext converted files were converted to the XML input format required by the pipeline, essentially by just wrapping the text in `<text>` and `<document>` elements and removing illegal characters.⁸ In run 2, however, the supplied HTML files were used, having been first run through an in-house HTML to XML converter.

PPI Extraction The named entity recognition and relation extraction stages of the pipeline (Section 2.1) were used to identify mentions of protein-protein interactions.

UniProt Normalisation Two approaches were used to assign UNIPROT identifiers to protein mentions, exact matching (in runs 1 and 2) and fuzzy matching (in run 3). In exact matching, the protein name in the text is compared against each protein synonym in the UNIPROT lexicon using a case-insensitive match, to obtain a list of possible identifiers. If no possible identifiers are found, and the protein name is the long or short form of an abbreviation identified by the abbreviation extractor, then the corresponding (short or long) form is also looked up in the lexicon. In order to filter the list of identifiers, each identifier is weighted according to how often its corresponding species name is mentioned in the text, with species name mentions closer to the protein mention receiving higher weights. The identifier with the highest weight is then chosen.

The fuzzy match protein normaliser uses a string distance measure (see Section 2.3) to find the set of protein names in the lexicon which are closest to the protein mention in the text. These distances are then weighted according to the species word mentions, as for exact matching, and the highest weighted identifier chosen.

Curation Filter The curation filter takes as its input the set of UNIPROT identifier pairs representing the interactions found in the text by the pipeline, with their UNIPROT normalisations, and outputs the set of normalised, curatable interactions. The filter was implemented with an SVM classifier (using SVM^{light} [7] with an RBF kernel), trained on the supplied training data, using the following set of features:

relation-count This feature counts the number of times that the interaction is mentioned in the document.

⁸These were ascii control characters inserted by `pdftotext`, which are not legal in XML. They were all removed except for ascii 0x0C, which was converted to a double newline.

inter-sentential This indicates whether the majority of the mentions of the interaction are inter-sentential relations between proteins, or intra-sentential. As noted in Section 2.1, the relation extractor does not perform well on inter-sentential relations, so very few of these are predicted (only 15 in the training corpus).

relation-confidence Each interaction mention found by the pipeline has an associated confidence. The value of this feature is the maximum confidence assigned to an interaction’s mentions.

position This feature specifies the relative position within the document of the first and last mentions of the interaction. In addition, the mean relative position of the interaction mentions is included, for each interaction.

species The species feature indicates whether the proteins in the proposed interaction have different species.

title This feature indicates whether the interaction is mentioned in the title.

normalisation-confidence When using the fuzzy-matched normalisations, this feature indicates how close a match has been found during normalisation of the protein mention.

As recommended in the IP Subtask instructions, any documents containing more than 30 interactions were excluded from the training set.

2.6 Interaction Sentence Subtask

The T6 Interaction Sentence Subtask system was identical to the system used for run 2 of the IP Subtask (see Section 2.5) with the addition of the following two steps:

Data Preparation The HTML to XML converter preserved a mapping between the HTML text and the sentences in the converted XML file.

Passage Selection The interaction mentions for each curatable interaction were sorted according to the confidence values associated with each mention and the sentences associated with the top five mentions were returned as the relevant passages.

3 Analysis

3.1 Gene Mention

As reported in Section 2.2, two different techniques were used for the gene mention task, conditional random fields (CRF) and bidirectional maximum entropy markov models (BMEMM). Runs 1 and 3 both used CRF (with different settings of the Gaussian prior) whilst run 2 used BMEMM, with all three runs using the same feature set. The results are shown in Table 2. The distribution of scores on the test

Run	Method	Heldout			Test		
		Precision	Recall	F_1	Precision	Recall	F_1
1	CRF	0.8594	0.8211	0.8398	0.8697	0.8255	0.8470
2	BMEMM	0.8597	0.7982	0.8278	0.8638	0.8041	0.8329
3	CRF	0.8463	0.8297	0.8379	0.8649	0.8248	0.8444

Table 2: Performance comparison for the Gene Mention task. In the heldout configuration, the system was trained on 80% of the data and tested on 20%, whilst in the test configuration the system was trained on all the training data and tested on the test data.

data matches that obtained during heldout testing on the training data, in that CRF outperformed BMEMM on F_1 (mainly due to higher recall) and the run 1 configuration was the best overall.

3.2 Gene Normalisation

As described in Section 2.3, we produced 3 runs for the Gene Normalisation task. The results are shown in Table 3.

Run	Method	Precision	Recall	F_1
1	ML Filter 1	0.767	0.601	0.674
2	ML Filter 2	0.767	0.606	0.677
3	Heuristics Filter	0.597	0.782	0.677

Table 3: Performance comparison for the Gene Normalisation task on the test data. The machine learning (ML) Filter 1 uses Dice measure to calculate the similarity between synonyms associated with the identifier and all entities detected by NER in the current document; while ML Filter 2 uses tf^*idf for the same task. The Heuristics Filter simply chooses the identifier that has the lowest number.

The approach is not completely supervised because the training data constructed for a document does not necessarily contain all the correct identifiers as given in the answer file. The coverage of our fuzzy matcher is up to 88%, which is an upperbound for the recall of the GN system. The approach takes advantage of string similarity measures that are more generic than hand-coded knowledge when carrying out the fuzzy matching. Combined with machine learning techniques, the T6 system is more portable than some GN systems reported in previous work [4, 3].

3.3 Interaction Article Subtask

Table 4 compares results of a bag-of-words baseline system to the bag-of-nlp system. The baseline system uses only the *word* and *bigram* features but is otherwise the same as the bag-of-nlp system. The results are presented both for 5-fold cross-validation on the training set and for the test set.

System	5-Fold cross-validation					Test				
	AUC	Prec	Rec	F_1	Acc	AUC	Prec	Rec	F_1	Acc
baseline	0.9757	0.9452	0.9420	0.9436	0.9276	0.8188	0.6898	0.8480	0.7608	0.7333
bag-of-nlp	0.9777	0.9550	0.9474	0.9512	0.9374	0.8483	0.6994	0.8747	0.7773	0.7493

Table 4: Overall Results

Data Inconsistency The most obvious observation is the drop in performance from cross-validation to test. This can be partially explained by some inconsistencies between the training and test sets. When analysing the test set, it was noticed that 37 of the files were actually also present in the training set. Furthermore, 13 of these files had a different label in the test set than in the training set: in each of the differences a document that was labelled as a positive example in the training set was labelled as a negative example in the test set. This would explain why the precision has gone down more than the recall. In order to estimate the effect of these differences, the bag-of-nlp system was trained on all of the training documents with the exception of these 13 documents and then used to predict the class of the 13 files. In 12 of the 13 cases, the system predicted that the articles were positive examples and thus found to be incorrect in the final evaluation. If these 13 files had been labelled as positive in the test set, the precision would have risen from the reported 0.699 to 0.725. A manual examination of some of the files in question suggests that the abstracts do contain interactions, but it is difficult to

determine if the full text versions meet the standards for curation. Regardless, the differences between the labels in test and training raise concerns about how representative the test data is of the training data.

NLP Benefits The next observation is that the bag-of-nlp system does provide a small improvement over the baseline system. The NLP features are based largely on either the NER module or the chunker. In order to assess the relative contribution of each component, a lesion test was performed where the system was run without NER and then without the chunker. The results are presented in Table 5.

System	5-Fold cross-validation					Test				
	AUC	Prec	Rec	F_1	Acc	AUC	Prec	Rec	F_1	Acc
bag-of-nlp	0.9777	0.9550	0.9474	0.9512	0.9374	0.8483	0.6994	0.8747	0.7773	0.7493
no NER	0.9771	0.9498	0.9465	0.9482	0.9334	0.8277	0.6908	0.8640	0.7678	0.7387
no chunker	0.9779	0.9530	0.9471	0.9501	0.9359	0.8412	0.6956	0.8773	0.7759	0.7467

Table 5: Benefits from NLP

The results indicate that the NER module is more useful than the chunker. Overall, however, the contribution from NLP is less significant than one would hope and less than reported in previous work [9]. One possibility is that since the baseline system already performs at a very high level, the contributions of imperfect NLP are not as effective. This is supported by the fact that the relation extraction component, which has an F_1 score of less than 0.50, actually hurt system performance and was therefore not included in the final bag-of-nlp system. In the future, it would be useful to perform experiments on a dataset that has been annotated with both document classes and linguistic information to determine the benefits of human-level NLP on document classification. This would at least provide an upper bound for how much improvement could be provided by NLP.

3.4 Interaction Pair Subtask

For the submissions to the Interaction Pair (IP) Subtask, exact matching normalisation was used in runs 1 and 2, and fuzzy matching in run 3, whilst the PDFconverted files were used in runs 1 and 3, and the HTML converted files in run 2. During cross-validation testing on the training set, the configuration in run 1 achieved the highest score, followed by the run 2 configuration, and then the run 3 configuration. However, as can be seen in Table 6, the run 3 configuration achieved the highest score on the test data.

Run	Filetype	Normaliser	10-fold cross validation			Test		
			Precision	Recall	F_1	Precision	Recall	F_1
1	PDF	exact	0.2687	0.1712	0.2091	0.2302	0.1283	0.1648
2	HTML	exact	0.2574	0.1702	0.2049	0.2003	0.1204	0.1504
3	PDF	fuzzy	0.2354	0.1756	0.2011	0.2131	0.1496	0.1758

Table 6: Comparison of performance for different data file types and normalisers. The system was tested using 10-fold cross-validation on the training data, and on the test data.

Since the overall system for the IP Subtask comprised several different stages, it would be useful to gain some idea of the performance of each stage to see where improvements could be made. In the rest of this section, each component will be considered in turn to discuss how it contributes to the overall IP Subtask errors.

Named Entity Recognition (NER) When tested on the TXM blind test corpus, the NER component achieves an F_1 score of 78% on protein mentions. Within the IP Subtask, NER can cause both false negatives, if the NER component does not correctly recognise a protein that is involved in a curatable interaction, and false positives, if the NER component incorrectly marks a non-protein as a protein, and that protein is then placed in an interaction and normalised by subsequent processing stages. The NER component can also make boundary errors, where it identifies a protein at the correct location but gets its boundaries wrong, making the task more difficult for the normaliser. There is no gold NER data available for the IP Subtask test documents, but an estimate of the recall of NER and normalisation combined can be obtained by counting the number of gold interactions in the IP Subtask test data where the system correctly identified and normalised both proteins in the interaction. Using the configuration in run 1 (exact match normaliser and pdf converted documents), both proteins were correctly identified in 43.86% of the gold interactions.

Relation Extraction (RE) The RE component, when tested on the TXM blind test corpus, using gold NE data, achieves an F_1 of about 45% on the identification of protein-protein interaction (PPI) mentions. Table 7 gives an upper bound on the recall of the RE component, in the context of the IP Subtask, by showing the counts of true positives obtained by considering all generated matches for all the protein pairs output by RE (note that the recall figure here is lower than the 43.86 mentioned in the previous paragraph, since the figures in the table only include those proteins which the RE component has predicted to be in interactions, whilst the 43.86 includes all proteins predicted by NER). The RE component can introduce false positives into the IP Subtask by identifying incorrect PPIs, which are then classed as curatable by the curation filter, and can introduce false negatives by missing mentions of curatable interactions. It is also possible that curatable interactions are not mentioned directly in the document, but are inferred from experimental descriptions, and so would never be detected by the RE component.

Normalisation In the normalisation component, a list of possible matches is generated for each protein mention using a string matching algorithm, and then this list of matches is reordered using the species information found in the text. The normalisation requirement in the IP Subtask complicated any error analysis, since the gold data (in the form of pairs of Uniprot identifiers) could not be matched directly with the text. Nevertheless, a measure of the recall of NER and normalisation combined was given above, and the effectiveness of the species-based disambiguation can be gauged from the results shown in Table 7. This table shows how the disambiguator reduces the number of false positives (obtained by pairing all matched normalisations for each predicted pair of interacting proteins) by about 3 orders of magnitude.

Curation Filtering Table 7 also illustrates the effectiveness of the machine learning based curation filter in removing false positives. In general it achieves around a 10-fold reduction in false positives, whilst removing around a third of true positives. The threshold of the filter could be adjusted to favour precision or recall, but for the IP Subtask submission it was optimised to give the highest possible F_1 when cross-validating on the training set.

In summary, the relation extractor and the normaliser seem to be the main areas where improvements could be made. The relation extractor achieves an F_1 of 45% on PPI mentions in the TXM data, which compares well to the inter-annotator agreement (IAA) of 52% on this data, but is low in absolute terms. It should be emphasised that this score is on PPI mentions, and since a curatable interaction may be mentioned several times, or perhaps not explicitly mentioned at all, it is not clear exactly what effect the score on PPI mentions has on the IP Subtask.

The low IAA was a cause for concern within the TXM project and thus efforts were made to improve it in the second round of annotation. This round was completed after the BioCreAtIVe II challenge

Filetype	Normaliser	Stage	TP	FP
PDF	exact	Generate matches	333 (29.46%)	1,121,979
		Disambiguate	223 (19.73%)	4,351
		Curation filter	145 (12.83%)	485
HTML	exact	Generate matches	314 (27.79%)	1,077,231
		Disambiguate	207 (18.32%)	3,939
		Curation filter	136 (12.04%)	543
PDF	fuzzy	Generate matches	449 (39.73%)	9,016,377
		Disambiguate	271 (23.98%)	8,069
		Curation filter	169 (14.96%)	624

Table 7: Comparison of performance on the IP Subtask test data before and after species-based disambiguation, and after curation filtering. The percentage of true positives (TP) is measured against the total number of gold interactions.

ended and employed several iterations of piloting the annotation and revising the guidelines before starting the annotation for real. The IAA on PPIs increased to an F_1 of 64.77%, a score which is still lower than might be hoped, but which is believed to accurately reflect the inherent difficulty of the task. Unfortunately, to the best of our knowledge, there are few published IAA figures from similar annotation tasks from other groups, making comparison with other work difficult.

As noted in Section 2.3, normalisation of proteins in biomedical text is a hard task, and the normalisation within the IP Subtask is especially hard as the species is not given in advance. From Table 7 it can be seen that disambiguation is a significant problem in normalisation, with up to 40% of correctly normalised pairs erroneously removed by the disambiguator.

3.5 Interaction Sentence Subtask

The preliminary results for the Interaction Sentence (IS) Subtask are shown in Table 8. As mentioned

Description	Value
No. eval. predicted passages	2,497
No. eval. unique passages	2,072
No. eval. matches to previously selected	147
No. eval. unique matches to previously selected	117
Fraction correct (best) from predicted passages	0.0589
Fraction correct (best) from unique passages	0.0565
Mean reciprocal rank of correct passages	0.5525

Table 8: IS Subtask Evaluation Summary

in the methods section, the passages selected for this system were derived from the output of run 2 of the IP Subtask system. The biggest drawback of this system is that the relation extraction module is trained to identify all protein-protein interactions and not just curatable interactions. Therefore, the confidences that are used to rank the passages do not take into account the curatability of the sentence, only the degree of certainty as to whether they represent a protein-protein interaction. It would be possible in the future to rerank these passages based on the training data provided as part of the ISS task.

A further drawback is that the IP Subtask system was optimised to correctly normalise the protein mentions. However, for the IS Subtask, it was not critical to identify the correct normalisations, but

rather just the correct passages. Thus, the system could potentially be improved by skipping the disambiguation step. Table 7 indicates that more than 100 correct interactions, over 30% of the total, were incorrectly filtered out during the disambiguation stage. Though it is difficult to determine how removing this stage would effect the IS Subtask scores, it does suggest that some improvement could be made.

4 Conclusions

For the PPI subtasks (IP, IS and IA), the information extraction pipeline developed for the TXM programme proved effective since it addressed related problems (identification of proteins and their interactions) and was trained on similar data to that used in BioCreAtIVe II. For the IP Subtask, the pipeline architecture was easily extended with two extra components (normalisation and curation filtering) specific to the requirements of the subtask, showing the flexibility of this architecture.

The approach to normalisation that we have adopted, based on a string distance measure and machine learning disambiguation, has the advantage that it should be more easily adaptable to other normalisation problems (e.g., tissues, cell-lines) than an approach based on manually created matching rules. Although better results may currently be obtained with rule-based methods, we believe that our proposed approach offers more promise for the future. Given that it is very hard to automatically predict the single correct identifier for a biomedical entity (such as a protein), it would be interesting to consider the relative merits of an approach which focuses on reducing the choice of identifiers to a small number, as compared to supplying the user with fuzzy matching tools to help search ontologies interactively.

The T6 approach to the IP Subtask involved trying to reconstruct curated information from interactions mentioned explicitly in the text; however, it is not known what proportion of curated data can be obtained this way. In other words, are all curated interactions mentioned explicitly as an interaction between two mentioned proteins? A recent paper [17] showed that a significant proportion of facts in the MUC evaluations are distributed across several sentences, and similar results may apply in the biomedical domain. While the low overall scores in the IP Subtask show that NLP techniques are not yet ready to replace manual curation, they may be able to aid curators in their work, or be used to produce large volume, noisy data which is of benefit to biologists.

5 Acknowledgements

The TXM pipeline on which this system was based was carried out as part of a joint project with Cognia (<http://www.cognia.com>), supported by the Text Mining Programme of ITI Life Sciences Scotland (<http://www.itilifesciences.com>). The authors would also like to thank Beatrice Alex, Mijail Kabadzov and Stuart Roebuck for all their assistance during the development of the system and the preparation of this manuscript.

References

- [1] N. Collier and K. Takeuchi. Comparison of character-level and part of speech features for name recognition in biomedical texts. *Journal of Biomedical Informatics*, 37(6):423–435, 2004.
 - [2] J. R. Curran and S. Clark. Language independent NER using a maximum entropy tagger. In *Proceedings of CoNLL-2003*, pages 164–167, 2003.
 - [3] H. Fang, K. Murphy, Y. Jin, J. S. Kim, and P. S. White. Human gene name normalization using text matching with automatically extracted synonym dictionaries. In *Proceedings of BioNLP’06*, New York, USA, 2006.

- [4] D. Hanisch, K. Fundel, H.-T. Mevissen, R. Zimmer, and J. Fluck. ProMiner: Organism-specific protein name detection using approximate string matching. *BMC Bioinformatics*, 6(Suppl 1):S14, 2005.
- [5] M. A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, 84:414–420, 1989.
- [6] M. A. Jaro. Probabilistic linkage of large public health data files. *Statistical in Medicine*, 14:491–498, 1995.
- [7] T. Joachims. Making large-scale support vector machine learning practical. In A. S. B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1999.
- [8] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
- [9] M. Matthews. Improving biomedical text categorization with NLP. In *Proceedings of the SIGs, The Joint BioLINK-Bio-Ontologies Meeting, ISMB 2006*, pages 93–96, 2006.
- [10] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 188–191. Edmonton, Canada, 2003.
- [11] R. McDonald and F. Pereira. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, 6(Suppl1):S6, 2005.
- [12] G. Minnen, J. Carroll, and D. Pearce. Robust, applied morphological generation. In *Proceedings of 1st International Natural Language Generation Conference (INLG'2000)*, 2000.
- [13] L. A. Nielsen. Extracting protein-protein interactions using simple contextual features. In *Proceedings of the BioNLP workshop, HLT/NAACL 2006 - poster session*, pages 120–121, 2006.
- [14] A. Schwartz and M. Hearst. Identifying abbreviation definitions in biomedical text. In *Pacific Symposium on Biocomputing*, pages 451–462, 2003.
- [15] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In M. Hearst and M. Ostellendorf, editors, *Proceedings of HTL-NAACL-2003*, pages 213–220, Edmonton, Canada, 2003.
- [16] L. Smith, T. Rindflesch, and W. J. Wilbur. MedPost: a part-of-speech tagger for biomedical text. *Bioinformatics*, 20(14):2320–2321, 2004.
- [17] M. Stevenson. Fact distribution in information extraction. *Language Resources and Evaluation*, 40(2):183–201, 2006.
- [18] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147, 2003.
- [19] Y. Tsuruoka and J. Tsujii. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 467–474, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.
- [20] W. E. Winkler. The state of record linkage and current research problems. Technical report, Statistics of Income Division, Internal Revenue Service Publication R99/04, 1999.



Extracting Interacting Protein Pairs and Evidence Sentences by using Dependency Parsing and Machine Learning Techniques

Güneş Erkan¹

gerkan@umich.edu

Arzucan Özgür¹

ozgur@umich.edu

Dragomir R. Radev^{1,2}

radev@umich.edu

¹ Department of Electrical Engineering and Computer Science, University of Michigan,
Ann Arbor, MI 48109, USA

² School of Information, University of Michigan, Ann Arbor, MI 48109, USA

Abstract

The biomedical literature is growing rapidly. This increases the need for developing text mining techniques to automatically extract biologically important information such as protein-protein interactions from free texts. Besides identifying an interaction and the interacting pair of proteins, it is also important to extract from the full text the most relevant sentences describing that interaction. These issues were addressed in the BioCreAtIvE II (Critical Assessment for Information Extraction in Biology) challenge evaluation as sub-tasks under the protein-protein interaction extraction (PPI) task. We present our approach of using dependency parsing and machine learning techniques to identify interacting protein pairs from full text articles (Protein Interaction Pairs Sub-task 2 (IPS)) and extracting the most relevant sentences that describe their interaction (Protein Interaction Sentences Sub-task 3 (ISS)).

1 Introduction

Protein-protein interactions play important roles in vital processes such as cell cycle control, and metabolic and signaling pathways. There are a number of (mostly manually curated) databases such as MINT [11] and SwissProt [1] that store protein interaction information in structured and standard formats. However, the amount of biomedical literature regarding protein interactions is increasing rapidly and it is difficult for interaction database curators to detect and curate protein interaction information manually. Thus, most of the protein interaction information remains hidden in the text of the papers in the biomedical literature. Therefore, the development of information extraction and text mining techniques for automatic extraction of protein interaction information from free texts has become an important research area.

There have been many approaches to extract protein interactions from free texts. One approach is matching pre-specified patterns and rules [2]. Although this approach achieves high precision, it suffers from low recall. The reason is that, cases which are not covered by the pre-defined patterns and rules can not be extracted. Another approach is using natural language processing (NLP) techniques such as full parsing [4] and partial parsing [8]. These parsing approaches consider sentence syntax only but not its semantics. Thus, although they are complicated and require many resources, their performance is not satisfactory. Machine learning techniques for extracting protein interaction information have gained interest in the recent years [5, 7, 10]. These studies usually use bag-of-words features, or only syntactic features extracted from sentences and do not consider any dependency or semantic information.

BioCreAtIvE II (Critical Assessment for Information Extraction in Biology) challenge evaluation¹ consists of three tasks, which are Gene Mention Tagging (GM), Gene Normalization (GN), and Protein Protein Interaction (PPI). We participated in two sub-tasks of PPI, Protein Interaction Pairs

¹http://biocreative.sourceforge.net/biocreative_2.html

Sub-task 2 (IPS) and Protein Interaction Sentences Sub-task 3 (ISS). In these subtasks participants were provided with a collection of 358 full text articles in HTML. The aim of IPS was to identify the interacting protein pairs in each article. The goal of ISS was to extract the most relevant sentences that describe an interaction between two given proteins. For each protein interaction pair, participants were required to return a ranked list of maximum 5 evidence passages describing their interaction. Here, we present our approach of using dependency parsing and machine learning techniques to handle these tasks. We extract features from the dependency parse trees of the sentences and use these features to train an SVM classifier to identify and rank sentences that describe an interaction. Dependency parse trees not only capture sentence syntax but also some of its semantics such as predicate-argument relationships. We also present the improved version of our system, where we extract paths between a protein pair in the dependency parse tree of a sentence and define two kernel functions for SVM based on the cosine and edit distance based similarities among these paths.

2 System Description

2.1 Pre-processing

In this step, the HTML articles are converted to plain text by using html2text tool². Next, tokenization is done such that each alphanumeric word and punctuation mark is considered as a separate token. Finally, articles are segmented into sentences by using the MxTerminator tool [9].

2.2 Protein Name Identification

In order to extract protein-protein interaction information from an article, first the protein names must be identified. For the BioCreAtIvE II challenge we were provided with a release of the SwissProt database [1]. We adapted the dictionary-based approach to identify protein names. We used the provided database as a dictionary to match the words in a sentence against the “description”, “gene name”, and “gene synonyms” fields. We preferred longer matches to shorter ones. If a sentence contains n different proteins, there are $\binom{n}{2}$ different pairs of proteins. Before parsing a sentence, we make multiple copies of it, one for each protein pair. To reduce data sparseness, we rename the proteins in the pair as *PROTX1* and *PROTX2*, and all the other proteins in the sentence as *PROTX0*.

2.3 Protein Name Conflict Resolution

For the BioCreAtIvE challenge we were supposed to output the UniProt IDs of the protein pairs that interact, not their names. Since proteins with the same name may have different UniProt IDs depending on their organism source, we use heuristics to identify the organism source of a protein and map a protein name to its corresponding UniProt ID. For each protein, we matched the candidate organism names and synonyms in the article and weighted them according to their proximity to the protein. In our weighting mechanism, the frequencies of the organism name appearing just before the protein name, in the same sentence with the protein name, and in the same article with the protein name are considered in descending order of importance. For instance, suppose we have a protein name “Alpha-adaptin A”. This protein has two candidate organism sources (human and mouse) and thus two candidate UniProt IDs (AP2A1_HUMAN and AP2A1_MOUSE) according to the UniProt table. We apply the following rule to select the correct UniProt ID of this protein.

1. Select the organism source that has the highest frequency of occurrence just before the protein name.
2. If it can not be disambiguated by (1), then select the organism source that has the highest frequency of occurrence in the same sentence with the protein name
3. If it can not be disambiguated by (1) and (2), then select the organism source that has the highest frequency of occurrence in the same article with the protein name

²<http://userpage.fu-berlin.de/~mbayer/tools/html2text.html>

4. If it can not be disambiguated by (1), (2), and (3), then select the organism source human (if one of the candidates is human)
 5. If it can not be disambiguated by (1), (2), (3), and (4), then the conflict can not be resolved.

2.4 Sentence Filtering

We assumed that protein-protein interaction sentences contain at least two proteins and an interaction word. Thus, we consider only such sentences and filter all the others. A list of interaction words, which consists of 45 noun and 53 verb roots, was compiled from the literature. We extended the list to contain all the inflected forms of the words and spelling variations such as *coactivate/co-activate* and *localize/localise*.

2.5 Feature Extraction with Dependency Parsing

Unlike constituent parsing, dependency parsing captures the semantic predicate-argument relationships in a sentence. We used the Stanford Parser³ to extract features from the dependency trees. For example, Figure 1 shows the dependency tree we got for the sentence “The results demonstrated that KaiC interacts rhythmically with KaiA, KaiB, and Sasa.” The final list of features used in our learning-based system is as follows.

- Each interaction word in our list is a binary feature by itself. In other words, if a particular interaction word occurs in a sentence, we set the corresponding feature for that sentence. If an interaction word is negated in the dependency tree, then we do not include that word, i.e. we assume that it does not occur in the sentence.
 - A binary feature that is set if the total distance of both protein names to an interaction verb in a sentence is 2, that is, if both protein names are the immediate children of an interaction verb.
 - An interaction verb that is an immediate parent of both proteins in the tree is a feature by itself.
 - The immediate parent node of each protein in the dependency tree is a feature by itself.
 - An interaction word that is an ancestor of a protein at one or two levels above it in the dependency tree is a feature by itself.

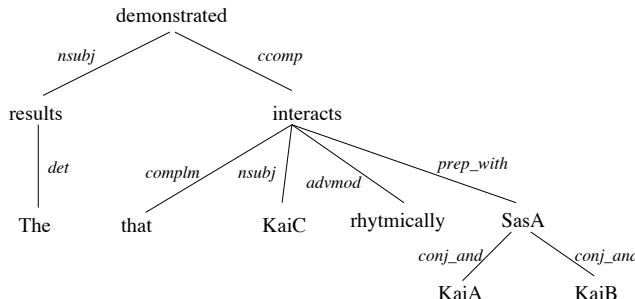


Figure 1: The dependency tree of the sentence “*The results demonstrated that KaiC interacts rhythmically with KaiA, KaiB, and SasA.*”

2.6 Machine Learning Techniques for Sentence Classification and Ranking

To classify sentences as containing an interaction or not and to rank the interaction sentences for each pair of proteins we used support vector machines (SVM). It has been shown to be one of the most powerful classifiers in general as well as the biomedical domain [5, 7, 10]. Our task is slightly different from that of the previous studies. Besides identifying protein-protein interaction pairs and sentences, we also identify the best sentences describing an interaction of a specific protein pair. We employ the strengths of both SVM and dependency parsing. We use the *SVM^{light}* library with linear kernel and default parameters [6] to classify sentences as containing an interaction or not and to rank the sentences that are classified as containing an interaction for each pair of interacting proteins.

³<http://nlp.stanford.edu/software/lex-parser.shtml>

As training set we used the Christine Brun corpus, provided as a resource by BioCreAtIvE. We first annotated the protein names in the corpus automatically and then, refined the annotation manually. As discussed in Section 2.2, each protein pair is marked as PROTX1 and PROTX2; and the remaining proteins in the sentence are marked as PROTX0. We ended up with 4,056 sentences. 2,704 of them are used for training and 1,352 for test. In the end, the system is trained with all the 4,056 sentences.

We ran the trained classifier on the test sentences and got a score (positive or negative) for each candidate sentence. To decide on whether we should output any sentence for a protein pair, we add up all the scores of the candidate sentences that contained the particular pair. If the sum for a pair is above a threshold, we output the top scoring (maximum of five) sentences for that pair and also output that pair as interacting. The threshold is set to 0 and 1 in runs 1 and 2, respectively. In run 3, we output the protein pair as containing an interaction, if there is at least one sentence that scored positive (IPS sub-task) and output the top 5 positive sentences for that pair (ISS sub-task).

2.7 Mapping Text to HTML

A requirement of the BioCreAtIvE challenge was that the predicted interaction sentences should come from the full text of the test set HTML articles. So, we had to map the extracted text sentences back to their HTML counterparts. We implemented an approximate string matching algorithm based on Levenshtein (edit) distance and an approximate token matching algorithm to handle this problem. First, we extracted the html passage where the sentence appears by using the approximate string matching algorithm. Next, we extracted the exact html sentence from that passage with the approximate token matching algorithm.

3 Improved System

Here, we present our improved system, where we use the shortest paths between a protein pair in the dependency parse tree of the sentence as features. We define two kernel functions for SVM based on the similarity between these paths. This system is developed after the submissions to the BioCreAtIvE challenge. Thus, all of our submitted runs used the system described in Section 2.

3.1 Sentence Similarity Based on Dependency Parsing

We define the similarity between two sentences based on the paths between two proteins in the dependency parse trees of the sentences. From the dependency parse trees of each sentence we extract the shortest path between a protein pair. For instance, in Figure 1 the path between *KaiC* and *SasA* is “*KaiC* - nsubj - interacts - prep_with - *SasA*”. Since, this sentence defines an interaction between *KaiC* and *SasA*, this is a positive instance. The path between *SasA* and *KaiA* is “*SasA* - conj_and - *KaiA*”. This sentence does not describe an interaction between *SasA* and *KaiA*. Thus, this path is a negative instance. If more than one path exists between the two proteins in a pair in the sentence (this may be the case if either of the proteins occurs more than once in the sentence), we select the shortest path. In our example sentence, there is a single path between each pair of proteins.

We define the similarity between two instances using cosine similarity and edit distance based similarity between the paths in the instances as follows.

3.1.1 Cosine Similarity

Suppose p_i and p_j are the paths between a protein pair in instance x_i and instance x_j , respectively. We represent p_i and p_j as vectors of term frequencies in the vector-space model. The cosine similarity measure is the cosine of the angle between these two vectors and is calculated as follows:

$$\text{cos_sim}(p_i, p_j) = \cos(\mathbf{p}_i, \mathbf{p}_j) = \frac{\mathbf{p}_i \bullet \mathbf{p}_j}{\|\mathbf{p}_i\| \|\mathbf{p}_j\|} \quad (1)$$

3.1.2 Similarity Based on Edit Distance

A shortcoming of cosine similarity is that it only takes into account the common terms, but does not consider their order in the path. For this reason, we also use a similarity measure based on edit distance (also called Levenshtein distance). Edit distance between two strings is the minimum number of operations that have to be performed to transform the first string to the second. In the original character-based edit distance there are three types of operations. These are insertion, deletion, or substitution of a single character. We modify the character-based edit distance into a word-based one, where the operations are defined as insertion, deletion, or substitution of a single word. We normalize edit distance by dividing it by the length (number of words) of the longer path, so that it takes values in the range [0, 1].

3.2 Kernel Function Definitions for SVM

We introduce two kernel functions for SVM based on the similarity functions that we defined in the previous sub-section. The cosine similarity based kernel K_{cos} and the edit distance based similarity kernel K_{edit} are defined as follows:

$$K_{cos}(x_i, x_j) = \text{cos_sim}(x_i, x_j); \quad K_{edit}(x_i, x_j) = e^{-\gamma \times \text{edit_distance}(x_1, x_2)} \quad (2)$$

The parameter γ is a positive number that allows us to tune K_{edit} to be symmetric positive definite, i.e., a well-defined kernel function.

4 Results and Discussion

Table 1 shows the summary of our results compared to the results obtained by the median system in the IPS sub-task. Interaction Pairs is the performance on identifying the interacting protein pairs. Interactor Normalization is the performance on interactor protein - article associations. To compute the average scores (av.) the scores for each article are computed separately and then the average is taken. However, some articles contain a single interaction, while others contain many interactions. Thus, the overall scores are also presented. Here, we report the scores of our best run (Run 2), in terms of F-score in the Interaction Pairs - Overall evaluation. We report the results for the cases where, articles containing exclusively interaction pairs that can be normalized to SwissProt entries are considered. Our results are higher than the median system in terms of all the evaluation metrics. Table 2 shows the summary results of our best run (Run 3), in terms of MRR score compared to the average results of all participating teams in the ISS sub-task. # Pred (A) is number of all predicted passages, # Pred (U) is number of unique passages, # TP (A) is number of and % (A) is fraction of true positives out of all predictions, # TP (U) is number of and % (U) fraction of true positives out of unique predictions, MRR is mean reciprocal rank of correct unique passages. The number of passages that we have predicted is higher than the average number of passages predicted by all the teams. Although the number of our true positives is much greater than the average, when we take the fraction over all the predictions it drops down. Our MRR score is slightly below the average MRR score reported by the BioCreative committee. However, they discuss that this score is not really statistically meaningful, as some teams didn't use the ranking system defined for this task. In their runs, each team was required to submit a ranked list of maximum 5 evidence passages for each interacting protein pair in each article. However, some teams ranked all the submitted passages as 1 in their runs, and some performed the ranking not for each pair but for the whole article. The BioCreative committee state that, when such runs are excluded from the statistics, the average MRR drops.

There is a lot of room for improvement in our system for both the IPS and ISS tasks. In the previous section, we discussed an improved version of our system, where we define kernel functions for

SVM based on the edit distance and cosine similarity among the paths between a protein pair in the dependency parse trees of the sentences. Our experiments with the Christine Brun corpus, show this system performs better in classifying a protein pair as interacting or not. However, all the other steps in the pipeline such as protein name identification, source organism disambiguation, and mapping text sentences back to html have an important affect on the overall performance and can be improved considerably.

Table 1: Summary of results of the IPS sub-task

Evaluation	Precision	Recall	F-score
Interaction Pairs - Overall (our results)	0.0759	0.1285	0.0954
Interaction Pairs - Overall (median)	0.0649	0.1179	0.0769
Interaction Pairs - Av. (our results)	0.0940	0.1988	0.0978
Interaction Pairs - Av. (median)	0.0808	0.2156	0.0842
Interactor Normalization - Overall (our results)	0.1478	0.3036	0.1988
Interactor Normalization - Overall (median)	0.1337	0.2723	0.1683
Interactor Normalization - Av. (our results)	0.2122	0.3269	0.2331
Interactor Normalization - Av. (median)	0.1707	0.3060	0.1922

Table 2: Summary of results of the ISS sub-task

Team	# Pred (A)	# TP (A)	# Pred (U)	# TP (U)	% (A)	% (U)	MRR
Our results	8355	5172	290	163	0.0347	0.0315	0.5329
Av. of all teams	6213.53846	3429.65385	207.46154	128.61538	0.04727	0.04725	0.55737

References

- [1] Bairoch, A. and Apweiler, R. (2000). The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucleic Acids Research*, **28**(1), 45–48.
- [2] Blaschke, C., Andrade, M. A., Ouzounis, C. A., and Valencia, A. (1999). Automatic extraction of biological information from scientific text: Protein-protein interactions. In *Proceedings of the AAAI Conference on Intelligent Systems for Molecular Biology (ISMb 1999)*, pages 60–67.
- [3] Bunescu, R., Ge, R., Kate, J. R., Marcotte, M. E., Mooney, R. J., Raman, K. A., and Wong, W. Y. (2005). Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, **33**(2), 139–155.
- [4] Daraselia, N., Yuryev, A., Egorov, S., Novichkova, S., Nikitin, A., and Mazo, I. (2004). Extracting human protein interactions from medline using a full-sentence parser. *Bioinformatics*, **20**(5), 604–611.
- [5] Donaldson, I., Martin, J., de Bruijn, B., Wolting, C., Lay, V., Tuckam, B., Zhang, S., Baskin, B., Bader, G. D., Michalockova, K., Pawson, T., and Hogue, C. W. V. (2003). Prebind and textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, **4**, 11. / *Bioinformatics*, **17**(Suppl 1), S97–S106.
- [6] Joachims, T. (1999). *Advances in Kernel Methods-Support Vector Learning*, chapter Making Large-Scale SVM Learning Practical. MIT-Press.
- [7] Mitsumori, T., Murata, M., Fukuda, Y., Doi, K., and Doi, H. (2006). Extracting protein-protein interaction information from biomedical text with svm. *IEICE Transactions on Information and Systems*, **E89-D**(8), 2464–2466.
- [8] Pustejovsky, J., Castano, J., Zhang, J., Kotecki, M., and Cochran, B. (2002). Robust relational parsing over biomedical literature: Extracting inhibit relations. In *Proceedings of the seventh Pacific Symposium on Biocomputing (PSB 2002)*, pages 362–373.
- [9] Reynar, J. C. and Ratnaparkhi, A. (1997). A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, D.C, USA.
- [10] Sugiyama, K., Hatano, K., Yoshikawa, M., and Uemura, S. (2003). Extracting information on protein-protein interactions from biological literature based on machine learning approaches. *Genome Informatics*, **14**, 699–700.
- [11] Zanzoni, A., Montecchi-Palazzi, L., Quondamatteo, M., Ausiello, G., Helmer-Citterich, M., and Cesareni, G. (2002). Mint: A molecular interaction database. *FEBS Letters*, **513**, 135–140.



Protein Interaction Sentence Identification by Using Hierarchical Pattern-Based Approach

Jung-Hsien Chiang¹
jchiang@mail.ncku.edu.tw

Tsung-Lu Michael Lee¹
michaellee@cad.csie.ncku.edu.tw

Yong-Xi Liu¹
liuyx@cad.csie.ncku.edu.tw

¹ Department of Computer Science and Information Engineering, National Cheng Kung University, 1, Da-Shuei Road, Tainan 701, Taiwan, ROC.

Abstract

Our system is a pattern-based architecture which identifies protein interaction patterns from biomedical literatures. The framework contains protein name recognition step, automated pattern generating step, pattern matching step, and sentence ranking step. The automated pattern generating step scans the positive interaction sentences and automatically constructs patterns based on the results of shallow parsing (chunking). A pattern must consist of at least one interaction keyword and two protein name entities. Our interaction keyword list includes 308 words with different tenses such as binding, binds, bind, and bound. Moreover, the patterns are built into three levels. From bottom up, the patterns go from specific to more general.

In the automated pattern generating step, hierarchical patterns are computed automatically with selected interaction key words and protein name entities. The sentence ranking procedure ranks each sentence according to the level of matched patterns and the confidence scores of interaction keywords. The hierarchical patterns provide different confidence levels (scores) that can be used to rank our sentences.

Keywords: text mining, information retrieval, protein-protein interactions, bioinformatics

1 Introduction

Proteomics and bioinformatics technologies have been widely used to analyze protein-protein interactions of complex biological systems, which are essential for understanding the mechanisms of human and cancer biology. The largest data resource of protein-protein interactions is the PubMed literatures provided at the U.S. National Library of Medicine that includes over 16 million citations from MEDLINE and other life science journals [1]. In recent years, the task of extracting protein-protein interaction from biomedical literatures has become one of the most challenging topics in the area of Bioinformatics. First of all, it is facing the challenge of protein name identification and normalization. Second, co-occurrence of any protein pairs in a sentence could generate many false-positive results.

In the last few years, pattern-based approach has been applied to tackle the challenge of protein interaction sentence identification [2,3]. Dynamic programming algorithm and key verbs are used to construct pattern templates [2]. A minimum description length (MDL)-based pattern-optimization algorithm is designed to optimize patterns [3]. We follow similar approach and implement an automated pattern generating system which can construct interaction pattern templates automatically from positive interaction sentences. The level of patterns is hierarchical and the level of patterns indicates different level of confidence of interaction sentence candidates.

In the protein name recognition step, we employ a java NLP tool, LingPipe [4], from BioCreAtIVe resources. Additionally, we apply a protein name dictionary from UniProt database [5] which contains

protein names, synonyms, and name descriptions. The matching process is implemented by using dynamic programming and modified alignment algorithm. Threshold values are required to identify the best matching protein name entities. The ambiguity of protein name entities is handled by using Entrez Gene database (gene2pubmed table) [6].

2 Method and Results

In this research, we implement three key modules (Figure 1) to handle the task of identifying protein interaction sentences or protein interacting pairs from full-text PubMed literatures. There are **Automated Pattern Generating** module, **Pattern Matching** module and **Sentence Ranking** module.

2.1 System Architecture

The system architecture shown in Figure 1 provides an overview of our proposed approach to identify protein interaction sentences or protein interacting pairs from full-text PubMed literatures. First, pattern templates are generated automatically from positive PPI sentences given by BioCreAtIVe resources at the **Automated Pattern Generating** module. Second, candidate sentences (those with at least two protein names) are evaluated with the pattern templates at the **Pattern Matching** module. Third, the matched sentences are ranked according to their interaction keywords, the level of pattern templates, and the gap between protein pairs at the **Sentence Ranking** module.

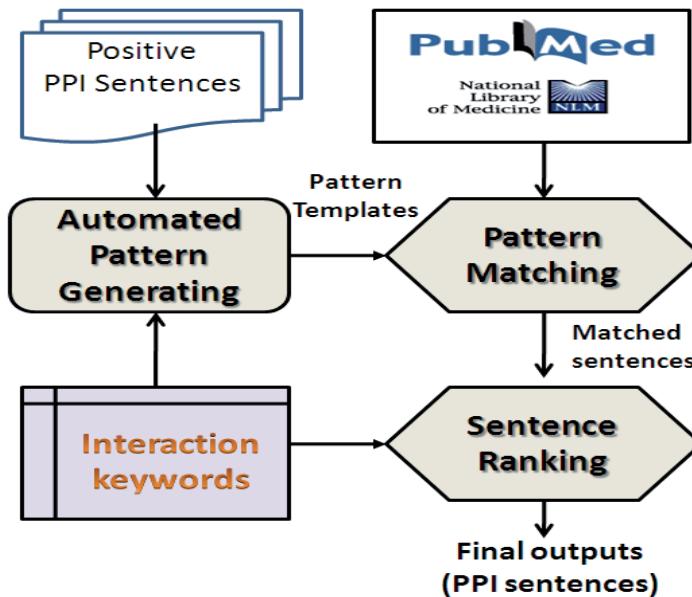


Figure 1: System Architecture.

2.2 Automated Pattern Generating Module

The predefined pattern templates are generated automatically in this module. The positive PPI sentences are collected from BioCreAtIVe resources. Sentences are first identified all the protein name entities and a chunking procedure is performed after NER process. For each positive PPI sentences, the system automatically identifies interaction keywords which appear in the sentences and two protein entities. To simplify the process and eliminate possible false positives, we did not use those sentences with more than two protein entities to generate pattern templates. For example, if a sentence has the following phrase “protein A interact with protein B” in it. The pattern generating module will automatically create a pattern “<P> interact with <P>” at the most specific level and “<P> interact <CC> <P>” at the higher level. Finally,

the pattern “<P> interact <P>” is constructed at the most general level. The purpose of this hierarchical pattern generating procedure is to handle variant types of interaction sentences. In addition, we give different confidence scores to rank each matched sentence according the level of matched patterns.

2.3 Pattern Matching Module

The pattern matching module is designed to match candidate sentences with pre-generated pattern templates. A sentence with the following phrase “protein C interact to a modified form of protein D” can only match to the most general pattern “<P> interact <P>” while the sentence with the phrase “protein E interact to protein F” can match to second level pattern “<P> interact <CC> <P>.” In other words, a candidate sentence is assigned a higher confidence score if it is matched to a more specific level of pattern templates.

The restriction to match the more specific level of pattern templates is higher than the general level. The hierarchical pattern matching procedure provides different level of confidence to evaluate all different types of sentence structures and semantics. Traditional rule-based pattern templates are fixed and one rule can only match one type of sentence structure. Our pattern matching module gives higher flexibility to users and at the same time maintains great consistency by using general to specific pattern levels.

2.4 Sentence Ranking Module

The last section of our proposed system is a sentence ranking module. All the matched sentences are ranked and sorted based on their matched pattern levels and on their interaction keywords identified in the sentence. We rank 308 interaction keywords from one to ten. Interaction keywords, such as, binding, interact, and inhibit are assigned to score ten. The final outputs are the sentences sorted by the sum of their pattern level scores and interaction keywords scores.

3 Discussions

Our hierarchical pattern-based framework is designed to improve traditional pattern-based architecture. Hierarchical pattern templates are generated automatically to handle variant types of sentence structure and grammar. The hierarchical pattern matching algorithm gives higher flexibility than traditional rule-based approach and maintains high precision by using most specific pattern level.

In the future work, we aim to analyze semantic structure of the sentences and automatically compute hierarchical semantic pattern-based algorithm to distinguish different interaction types according to their semantic meanings.

References

- [1] <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed>
 - [2] Huang, M., Zhu, X., Hao, Y., Payan, D.G., Qu, K., and Li, M., Discovering patterns to extract protein–protein interactions from full texts, *Bioinformatics*, 20: 3604-3612, 2004.
 - [3] Hao, Y., Zhu, X., Huang, M., and Li, M., Discovering patterns to extract protein–protein interactions from the literature: Part II, *Bioinformatics*, 21: 3294-3300, 2005.
 - [4] <http://www.alias-i.com/lingpipe/>
 - [5] Wu, C.H., Apweiler, R., Bairoch, A., Natale, D.A., *et al.*, The Universal Protein Resource (UniProt): an expanding universe of protein information, *Nucleic Acids Res.*, 34: D187-191, 2006.
 - [6] <http://www.ncbi.nlm.nih.gov/Entrez/>



BioText Report for the Second BioCreAtIvE Challenge

Preslav Nakov¹

nakov@cs.berkeley.edu

Anna Divoli²

divoli@ischool.berkeley.edu

¹ EECS, CS division, University of California, Berkeley, CA 94720

² School of Information, University of California, Berkeley, CA 94720

Abstract

This report describes the BioText team participation in the Second BioCreAtIvE Challenge. We focused on the Interaction-Article (IAS) and the Interaction-Pair (IPS) Sub-Tasks, which ask for the identification of protein interaction information in abstracts, and the extraction of interacting protein pairs from full text documents, respectively. We identified and normalized protein names and then used an ensemble of Naive Bayes classifiers in order to decide whether protein interaction information is present in a given abstract (for IAS) or a pair of co-occurring genes interact (for IPS). Since the recognition and normalization of genes and proteins were critical components of our approach, we participated in the Gene Mention (GM) and Gene Normalization (GN) tasks as well, in order to evaluate the performance of these components in isolation. For these tasks we used a previously developed in-house tool, based on database-derived gazetteers and approximate string matching, which we augmented with a document-centered ambiguity resolution, but did not train or tune on the training data for GN and GM.

Keywords: protein-protein interaction, gene/protein name recognition and normalization, ensemble of classifiers.

1 Introduction

The BioText team participated in the following tasks and sub-tasks of the Second BioCreAtIvE Challenge:

- *Gene Mention (GM) Task*
- *Gene Normalization (GN) Task*
- Protein-Protein Interaction:
 - *Interaction-Article Sub-Task (IAS)*
 - *Interaction-Pair Sub-Task (IPS)*

Our main interest and focus were the protein-protein interaction sub-tasks; however, since our method required the recognition and normalization of gene/protein¹ name mentions in the text, we also submitted runs for the GM and GN tasks in order to evaluate the performance of these components in isolation.

For the GM and GN tasks we adapted an in-house tool (without further training), which uses a gazetteer and expansion rules, and for the IAS and IPS we trained a number of Naive Bayes classifiers using various features. The following sections present each task/sub-task separately, explain in detail the applied method and discuss the results.

¹Since gene names and protein names are often interchangeable, below, when we refer to *gene names* (in GM and GN tasks) or *protein names* (in IAS and IPS sub-tasks), we implicitly mean *gene and/or protein names*.

2 Gene Mention Task (GM)

Given a sentence, the GM task asks the participants to return a list of the mentioned gene names. We address the problem by combining an EntrezGene-derived gazetteer with a rule-based approximate string matching algorithm.

2.1 Method

We used an in-house gene recognition and normalization tool, originally developed for the TREC 2003 Genomics Track [1] and extended for this year's BioCreAtIVe.

The original tool identified gene/protein names in raw text and mapped them to one or more LocusLink unique identifiers. The tool's gazetteer was limited to gene/protein names and their known synonyms listed in LocusLink, which were further filtered using WordNet [2] in order to remove common words like *or*, *and*, etc., which can be also gene names.

The original tool used a set of normalization and expansion rules in order to allow for some variations in form. These rules include token rearrangement, as well as removal of whitespace, commas, parentheses and numerals. All possible normalizations and expansions of all known LocusLink gene/protein names and their synonyms were generated off-line and then matched against a normalized version of the input text, giving priority to longer matches. The matches were then mapped back to the original text, and the corresponding IDs were assigned.

For our BioCreAtIVe participation, we significantly modified this tool. First, we downloaded the latest version of EntrezGene (which supersedes LocusLink) and extracted the IDs and the corresponding fields likely to contain variations of gene names, e.g. *name*, *official name*, *official symbol*, *alias* and *description*. We also made a clear separation between normalization and expansion rules, splitting the latter into two sub-groups: *strong rules* and *weak rules*, according to our confidence that the resulting transformation reflects the original names/synonyms. The strong rules allow only minor changes like:

- removal of white space (e.g., “*BCL 2*” → “*BCL2*”)
- substitution of non-alpha-numerical characters with a space (e.g. “*BCL-2*” → “*BCL2*”)
- concatenation of numbers to the preceding token (e.g., “*BCL 2*” → “*BCL2*”).

The weak rules remove at least one alpha-numeric token from the string. An example weak rule is the removal of trailing numbers e.g., “*BCL 2*” → “*BCL*”. As another example, treating a “/” as a disjunction produces two new strings:

$$\text{“aspartyl/asparaginyl beta-hydroxylase”} \rightarrow \begin{aligned} &\text{“aspartyl beta-hydroxylase” or} \\ &\text{“asparaginyl beta-hydroxylase”} \end{aligned}$$

Another weak rule handles parenthesized expressions, removing text before, within and/or after the parentheses. For example,

$$\text{“mitogen-activated protein (MAP) kinase”} \rightarrow \begin{aligned} &\text{“mitogen-activated protein (MAP)”}, \text{ or} \\ &\text{“mitogen-activated protein kinase”}, \text{ or} \\ &\text{“MAP kinase”}, \text{ or} \\ &\text{“mitogen-activated protein”}, \text{ or} \\ &\text{“MAP”}, \text{ or} \\ &\text{“kinase”}. \end{aligned}$$

Unlike in the original tool, the new rules have no priorities and are applied in parallel and recursively, trying all feasible sequences. For each resulting expanded variant, we record the ID of the source gene/protein/synonym and whether a weak rule was used at least once during its derivation. For a given variant, there are multiple possible IDs, some of which use strong rules only and others that use at least one weak rule. The strong variants are meant to be very accurate, while the weak ones are good for recall enhancement.

2.2 Runs

We have submitted three runs, which differ by the following two parameters:

- whether weak rules are used or not;
- whether the tool is allowed to use synonyms from the description field in EntrezGene.

The description field in EntrezGene often contains additional gene/protein synonyms, but can contain other things as well, e.g. chemicals, organism names, etc. Therefore it is a good source for recall enhancement at the expense of precision.

The first run targets precision, while the other two are recall-oriented.

- **Run 1**

No weak rules; no synonyms from the description field.

- **Run 2**

No weak rules; uses synonyms from the description field.

- **Run 3**

Uses weak rules; uses synonyms from the description field.

2.3 Results and Analysis

The results for our submissions for the GM task are shown in Table 1. As expected, both adding synonyms from the description field and using weak rules lead to dramatic increase in recall at the expense of precision. Our best F-score (62.29%) is achieved by Run 2, which is a compromise: it uses the description field, but no weak rules.

Run	P	R	F
1	61.53	58.82	60.15
2	60.56	64.11	62.29
3	54.13	68.22	60.36

Table 1: GM Results (in %).

3 Human Gene/Protein Name Normalization Task (GN)

Given an abstract, the GN task asks the participants to return a list of the EntrezGene identifiers and corresponding text excerpts for each mentioned human gene or gene product. We addressed the problem by combining a rule-based approximate string matching approach with a document-centered ambiguity resolution algorithm.

3.1 Method

We participated with the same gene recognition and normalization tool, we used for the GM task, adapting it for the normalization task by restricting it to the master list of human gene/protein IDs (as provided by the organizers for that task) and by using strong rules only.

The major problem was ambiguity. For example, *SYT* can refer to two human genes whose IDs are in the master list, *SYT1* (ID 6857) and *SS18* (ID 6760), and we need to choose one of them. For this purpose, we adopted a document-centered disambiguation approach, which has been successfully applied to text normalization [3] and word sense disambiguation [6]. In the case of word sense disambiguation, this is reduced to two principles: (1) *one sense per collocation* (i.e. assign a single ID for each gene/protein instance); and (2) *one sense per discourse* (assign the same ID to all instances of a given gene/protein within a document).

We add a third weak principle: (3) *no synonyms*. It assumes that, as a general preference, in case multiple names are possible in the literature for a given gene/protein name, in a particular document, authors tend to stick to just one of them. This means that two different gene names are unlikely to refer to the same gene/protein ID in the same text. One notable exception is when the gene/protein is mentioned for the first time in the text, in which case authors are likely to introduce some synonyms, typically the correspondence between the full name and the abbreviation they will use throughout the rest of the text e.g. “The dopamine D₄ receptor gene (*DRD4*) shows considerable homology to *DRD2*.” At present, we are not trying to model this, but it could be done easily, by adding a gene/protein name expansion recognizer, e.g. the one described in [4].

We support a set of possible IDs for each gene/protein name instance in the text, and once we assign a particular ID to some gene/protein name, we remove it from the set of IDs of all the rest and we implement the following three-step algorithm:

- **Step 1:** Assign the IDs for all unambiguous gene/protein instances, i.e. the ones for which there is a single possible ID.
- **Step 2:**
 1. Exclude all IDs recognized so far from all lists of possible candidates.
 2. Assign the corresponding ID for all unambiguous gene/protein instances.
 3. If there was at least one new assignment, go to 1.
- **Step 3:**
 1. Exclude all IDs recognized so far from all lists of possible candidates.
 2. Assign the current instance an ID from the set of its currently available IDs.
 3. If there was at least one new assignment, go to 1.

On Step 2, we consider the instances sorted by length in descending order (i.e. we prefer to cope with the long forms first), while on Step 3, we sort them by $(1/I + 0.001 \times L)$, where I is the number of different possible IDs for that instance, and L is the instance length (i.e. we prefer less ambiguous instances, and among the ones with the same level of ambiguity, we prefer the longer ones).

3.2 Runs

We submitted three runs:

- **Run 1:** step 1 only;
- **Run 2:** steps 1 and 2;

- **Run 3:** all three steps.

The first run targets precision, while the other two are recall-oriented.

3.3 Results and Analysis

The results for our submissions for the GN task are shown in Table 2. The best run is Run 1 ($F=68.7\%$), but Run 2 is virtually indistinguishable from it ($F=68.4\%$). Run 3 has a little better recall, but loses a lot on precision and ends up with a much worse $F=63.7\%$. Further analysis is needed in order to find out whether the bad performance of run 3 is due to a frequent violation of our assumption (3) or is what is to be expected by chance: in step 3 we make a forced random choice from the IDs of the confusion set. If this set contains, for example, 5 IDs, then there is only 20% probability to make the correct choice. Finally, as our results for GM the task suggest, our gene/protein identifier is far from perfect and generates many false positives, in which case we have no correct choice to make on step 3.

Run	P	R	F
1	0.716	0.661	0.687
2	0.702	0.666	0.684
3	0.580	0.707	0.637

Table 2: GN Results

4 Protein Interaction Article (IAS)

For the IAS sub-task, given a set of PubMed abstracts, we were asked to decide for each one whether it contained information that is relevant for protein interaction annotation or not, and to produce two ranked lists of PMIDs: one positive and one negative. We used an ensemble of Naive Bayes classifiers, each of which decides whether the document is positive or negative. The classifiers' posterior probabilities were then combined in order to produce a ranking within each list (positive and negative).

4.1 Method

4.1.1 Features and Parameters

We considered a number of features to train our classifiers. We used the same recognition and normalization tool we employed for the GM and the GN tasks, in order to identify UniProt genes/proteins (which, in this report, we call *UniProt annotations*) in the abstracts. We used the same tool to recognize MeSH terms and their synonyms in the text (which we call *MeSH annotations*). We also retrieved the MeSH terms associated with each abstract in PubMed. Finally, we used the abstract's words: stop-list filtered and TF.IDF weighted.

In order to increase the flexibility of our system, we imposed some limitations (parameters) on the features. See Table 3 for details. For example, limiting to specific MeSH tree branches (LB) was an *ad-hoc* decision in order to take into account only terms that we consider likely to be associated with descriptions of protein interactions. Setting a limit on the length of the MeSH tree level (TL) takes advantage of the MeSH hierarchy and groups related terms together. Restricting the detection of UniProt and MeSH annotation to strong rules only (SRO) boosts precision at the expense of recall. Finally, control over the frequency of terms reduces the number of word-features considered and helps overcoming some computational limitations.

Features	Parameters
MeSH terms	Minimum frequency (MF). Limit to the following MeSH tree branches: A, B, C, D and G (LB). Limit on the maximum MeSH tree level (TL).
Word TF.IDF weights (after removal of stopwords)	Minimum frequency (MF). Limit to the following interaction words: <i>interact, bind, activate, inhibit</i> and <i>mediate</i> (IWO).
UniProt annotations	Minimum frequency (MF). Restrict to strong rules only for term recognition (SRO).
MeSH annotations	Minimum frequency (MF). Restrict to strong rules only for term recognition (SRO). Limit to the following MeSH tree branches: A, B, C, D and G (LB). Limit on the maximum MeSH tree level (TL).

Table 3: Features and parameters used for the IAS task.

4.1.2 Classification

Most models were trained on the positive and the negative training data, but some also used a quarter of the noisy data, which was considered positive. We only used a quarter, in order to keep the positive/negative ratio more balanced.

Due to memory limitations and inter-dependencies between the different kinds of features, we did not use them all in one model, but instead trained an ensemble of 15 independent Naive Bayes classifiers (as implemented in WEKA [5]), and then we then combined their posteriors. See Table 4 for details.

Model	Training Data	Features	Parameters
1	PN	Word TF.IDF weights	MF = 10
2	PN	Word TF.IDF weights	MF = 20
3	PN	Word TF.IDF weights	IWO
4	PN	MeSH terms	MF = 3, LB, TL = 3
5	PN	MeSH terms	MF = 5, LB, TL = 2
6	PN	MeSH terms	MF = 50, LB, TL = 2
7	PN	MeSH annotations	MF = 10, LB, TL = 3
8	PN	MeSH annotations	MF = 5, LB, TL = 2
9	PN	UniProt annotations	MF = 10, SRO
10	PNN	Word TF.IDF weights	MF = 10
11	PNN	Word TF.IDF weights	MF = 20
12	PNN	Word TF.IDF weights	IWO
13	PNN	MeSH terms	MF = 3, LB, TL = 3
14	PNN	MeSH terms	MF = 5, LB, TL = 2
15	PNN	MeSH terms	MF = 5, LB, TL = 2

Table 4: Models used for classification and ranking of abstracts. We use the following abbreviations: PN = positive and negative; PNN = positive, negative and noisy; MF = min frequency; IWO = interaction words only; LB = limited MeSH tree branches; TL = max tree level (e.g., if TL = 2, then the MeSH tree label is cut to 7 characters); SRO = strong rules only.

4.2 Runs

We submitted 3 runs, representing different ways of combining the posteriors of the 15 classifiers described in Table 4.

- Run 1:

The primary classifier was *model 1* (Table 4); its posterior was given a weight of 100, while each of the remaining 14 models were given a weight of 1. In addition, we adjusted the binary decision boundary so that the output reflects the positive/negative proportion in the training data.

- **Run 2:**

The primary classifier was *model 10* (Table 4), which differs from model 1 only because it is trained on noisy data as well. As for run 1, the primary model was given a weight of 100, while each of the other models were given a weight of 1. The decision boundary was adjusted as in run 1.

- **Run 3:**

The primary model was *model 13* (Table 4), and it was given a weight of 5/3. As before, the other models were given a weight of 1, and the decision boundary was adjusted as in runs 1 and 2.

4.3 Results and Analysis

Our submissions for this sub-task aimed to: (a) study the effect of using the “noisy” data for training, and (b) experiment with ensembles of classifiers and feature combinations.

Table 5 shows the results. A comparison of the first two runs shows that using “noisy” data on training degrades the performance. In the third run, where all models were considered more uniformly, the performance improved consistently on all measures: precision, recall, F-measure, accuracy and AUC.

Run	P	R	A	F	AUC
1	0.586	0.589	0.587	0.588	0.625
2	0.497	0.504	0.497	0.501	0.576
3	0.608	0.688	0.623	0.646	0.655

Table 5: **IAS Results:** precision (P), recall (R), accuracy (A), F-score (F), AUC

5 Protein Interaction Pairs (IPS)

For the IPS sub-task, given a set of full text articles, we were asked to produce for each one a ranked list of interacting UniProt IDs. We built a classifier, which, given a pair of UniProt IDs, from the same organism and co-occurring in the same sentence, decides on whether they interact or not.

5.1 Method

5.1.1 Protein Identification

We adapted the tool we used for the GM and GN tasks for the present sub-task by restricting it to the master list of UniProt IDs provided by the organizers. We used the tool for the recognition of the proteins in each sentence. We have limited it to strong rules only, and we accepted both proteins and genes (below we refer to both as *proteins*). We only considered sentences that contained at least two different proteins; we also had a limitation on the maximum number of proteins per sentence. Ambiguity was a major problem, as the same protein often had multiple different IDs. We tried to disambiguate within the sentence by restricting the possible IDs to ones from the same organism. We also preferred IDs from an organism that was mentioned in the document’s MEDLINE record.

5.1.2 Classification

We made the simplifying assumption that, if two proteins interact, there should be a sentence in which they co-occur and which describes the interaction. For each training document, we were given

a list L of interacting protein pairs². However, no sentences containing the interaction were provided; therefore, on training, we assumed that for any pair (x, y) in L , a sentence containing both x and y was a positive example. From the remaining sentences, we used as negative examples the ones containing at least two different proteins. We used a Naive Bayes classifier (as implemented in Weka [5]) with the following features:

- length of the first protein (in characters)
- length of the second protein (in characters)
- distance between the two proteins (in characters)
- distance between the two proteins (in tokens)
- number of other proteins between the two interacting ones
- total number of proteins in the sentence
- ratio of the sentence number and the total number of sentences in the document
- words (TF.IDF weighted; no stopwords) in the sentence

In order to limit the number of candidates and to keep a more balanced positive/negative ratio, we introduced some additional restrictions: minimum and maximum protein length (in characters), maximum number of characters between the interacting proteins, maximum number of different proteins in the sentence. We also required the accepted word features to be present in pre-specified minimum number of documents.

Metric	Run 1	Run 2	Run3
All Articles			
mean P	0.055	0.034	0.157
mean R	0.189	0.235	0.185
mean F	0.073	0.053	0.138
overall P	0.057	0.033	0.134
overall R	0.148	0.200	0.096
overall F	0.082	0.057	0.111
Articles with SwissProt Normalized Pairs			
mean P	0.062	0.037	0.165
mean R	0.215	0.253	0.196
mean F	0.082	0.056	0.147
overall P	0.063	0.036	0.142
overall R	0.166	0.216	0.105
overall F	0.092	0.061	0.121

Table 6: IPS Results: Detection of Normalized Interaction Pairs

²In fact, for each document, we were given sets of interacting proteins; for each such set, we generated all possible protein pairs.

5.2 Runs

- Run 1

We used *full text* from PDF2txt (both for training and testing). All features were used, and the parameters were adjusted as follows: the interacting proteins were required to be 3-12 characters long, up to 100 characters apart, and the only proteins in the target sentence. Words were accepted as features, only if they appeared in at least 10 different documents.

- Run 2

This run was more liberal. Again, we used *full text* from PDF2txt and all features. The parameters were adjusted as follows: the interacting proteins were required to be 3-12 characters long, up to 200 characters apart, and up to three different proteins were allowed in the target sentence. Words were accepted as features, only if they appeared in at least 20 different documents.

- Run 3

Our third run used *abstracts only* (both for training and testing). We considered all features, except for the one that looks for the sentence's position in the document. There were no other restrictions.

5.3 Results and Analysis

Our submissions for this sub-task aimed to: (a) compare full text with abstracts, and (b) experiment with different distances (in characters) between the interacting proteins.

The results are presented in Tables 6 and 7. Run 3, which used only abstracts, performed best in terms of P and F (but not R) across all evaluations. Both runs 1 and 2 used full text. Run 1 was more restrictive for distance and therefore achieved higher P but lower R compared to run 2. It also achieved higher F .

6 Discussion and Future Work

Our best performing run was on a protein-protein interactions sub-task: IPS, run 3 – the only (sub)task where we used organism filtering for gene/protein name disambiguation. We believe considering organisms would also have improved our results for GN and IAS, where the ambiguity of gene/protein IDs was a major problem; we plan extra experiments in order to test this hypothesis. We also want to study the impact of different features and better ways of combining them.

Surprisingly, the aforementioned IPS run 3 used abstracts only, instead of full text documents. This could be due to a number of reasons. It is possible that two proteins are more likely to interact, if they co-occur in an abstract rather than a full document sentence. It is also possible that an interaction mentioned in an abstract is more likely to make its way in databases of protein interactions (we trained our algorithm assuming only interactions listed in such databases are positive examples). We would like to look into this in more detail.

Finally, as our GM and GN evaluation results show, we need to improve our gene/protein recognizer and normalizer. The training/testing data from the GN and GM tasks would be very useful both for supporting error analysis and for parameter tuning.

We look forward to future BioCreAtIvE challenges. Despite the text mining difficulties full text documents present, they are a great resource, and we believe future bioscience journal search engines will be built on these rather than on PubMed abstracts.

Acknowledgements: This work was supported by NSF DBI-0317510 grant.

Metric	Run 1	Run 2	Run 3
All Articles			
Mean for all evaluated articles			
P	0.115	0.079	0.225
R	0.425	0.518	0.291
F	0.168	0.130	0.227
Mean for evaluated articles with predictions			
P	0.122	0.083	0.303
R	0.449	0.546	0.393
F	0.177	0.137	0.306
Overall for the SwissProt interactor proteins			
P	0.111	0.074	0.259
R	0.406	0.496	0.257
F	0.174	0.128	0.258
Articles with SwissProt Normalized Pairs			
Mean for all the evaluated articles			
P	0.130	0.085	0.247
R	0.460	0.536	0.316
F	0.188	0.139	0.252
Mean for evaluated articles with predictions			
P	0.140	0.091	0.322
R	0.495	0.574	0.419
F	0.202	0.149	0.329
Overall for the SwissProt interactor proteins			
P	0.083	0.053	0.195
R	0.442	0.520	0.282
F	0.140	0.096	0.231

Table 7: IPS Results: Detection of Normalized Interactor Proteins

References

- [1] Bhalotia, G., Nakov, P., Schwartz, A., and Hearst, M. Biotext team report for the TREC 2003 Genomics Track. In *Proceedings of TREC* (Gaithersburg, MD, 2004).
- [2] Fellbaum, C., Ed. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
- [3] Mikheev, A. Document centered approach to text normalization. In *Proceedings of SIGIR* (2000), pp. 136–143.
- [4] Schwartz, A., and Hearst, M. A simple algorithm for identifying abbreviation definitions in biomedical texts. In *Proceedings of Pacific Symposium on Biocomputing (PSB 2003)* (2003), pp. 136–143.
- [5] Witten, I. H., and Frank, E. *Data Mining: Practical machine learning tools and techniques*, 2 ed. Morgan Kaufmann, 2005.
- [6] Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL* (1995), pp. 189–196.



LingPipe for 99.99% Recall of Gene Mentions

Bob Carpenter
carp@alias-i.com

Alias-i, Inc., 181 North 11th St., Brooklyn, NY 11211, USA

Abstract

Text data mining over biomedical research literature is a needle-in-a-haystack problem. We contend that first-best methods performing at 90% F-measure are insufficient, especially given that performance is much worse for “unseen” phrases. In this paper, we recast the problem as one of n -best search rather than first-best database population. We describe LingPipe’s HMM and character language model-based chunkers, which extract mentions of genes in unseen MEDLINE abstracts at 99.99% recall with greater than 50% mean-average precision. We provide evaluation results in terms of received precision-recall curves on unseen data.

Keywords: named entity extraction, confidence ranking, text data mining, search, character language models, hidden Markov models, forward-backward algorithm, A* algorithm

1 Introduction

Using a first-best entity extractor is akin to removing Google’s “Search” button and relying on “I’m Feeling Lucky”. Even with state-of-the-art precision, recall is going to be unacceptable for individual research or data mining purposes, which are often of the needle-in-a-haystack variety. Researchers don’t need to find dozens or hundreds of references to a common pathway interaction, they need to find the rare references that link two of the genes that are differentially expressed in a series of microarray assays in an unexpected way.

Evaluations by F-measure overemphasize performance on common, oft-repeated mentions. When performance is reported on mentions not included in the training data, error rates typically double or more. The alternative we offer is n -best output with conditional probability estimates of the mention given the text. This normalizes scores across sentences and documents, allowing the annotation problem to be recast as a search problem. We believe that scoring metrics for search, such as average precision and area under the receiver operating characteristic curve or log loss, are more appropriate for evaluating real-world uses of text data mining than 0/1-loss (first-best).

LingPipe’s confidence-based chunkers are first-order hidden Markov models with emission probabilities estimated by (padded) character language models. Using a generalized form of best-first search over the lattice produced by the forward-backward algorithm, these chunkers are able to iterate an arbitrary number of chunks in confidence-ranked order. Setting the threshold to 99.999% recall, these chunkers run at 330,000 characters/second.

LingPipe also contains a longer-distance character-language-model based chunker that rescores n -best whole-sentence analyses from the confidence-based chunker. We submitted a run of that chunker to BioCreAtIVe, as well as confidence-based results. See [2] for a description of the rescored model.

2 LingPipe's Character Language Models

LingPipe's classification, tagging, and entity extraction are all based on n -gram character language models. Language models define probability distributions $p(\sigma)$ over strings $\sigma \in \Sigma^*$ drawn from a fixed alphabet of characters Σ . LingPipe adopts a standard random process approach to n -gram language models, where probabilities are normalized over strings of a fixed length.

The process models factor the probability $p(\sigma c)$ of the string σ followed by the character c using the chain rule: $p(\sigma c) = p(\sigma) \cdot p(c|\sigma)$. The n -gram Markov assumption restricts the context of a conditional estimate $p(c|\sigma)$ to the last $n - 1$ characters of σ , taking $p(c_n|\sigma c_1 \dots c_{n-1}) = p(c_n|c_1 \dots c_{n-1})$.

The maximum likelihood estimator for this model is $\hat{p}_{ml}(c|\sigma) = \text{count}(\sigma c) / \text{extCount}(\sigma)$, where $\text{count}(\sigma)$ is the raw corpus count of the string σ and $\text{extCount}(\sigma) = \sum_c \text{count}(\sigma c)$ is the number of single character extensions of σ .

LingPipe interpolates all orders of maximum likelihood estimates using Witten-Bell smoothing [4]. The smoothed estimates are defined by $\hat{p}(c|d\sigma) = \lambda(d\sigma)\hat{p}_{ml}(c|d\sigma) + (1 - \lambda(d\sigma))\hat{p}(c|\sigma)$ with the boundary condition $\hat{p}() = 1/\text{size}(\Sigma)$ given by the uniform distribution. Witten and Bell smoothing takes the interpolation ratio $\lambda(\sigma) = \text{extCount}(\sigma) / (\text{extCount}(\sigma) + \theta \cdot \text{numExts}(\sigma))$, where $\text{numExts}(\sigma) = \text{size}(\{c | \text{count}(\sigma c) > 0\})$. The free parameter θ , which controls the degree of smoothing, was fixed at 1.0 by Witten and Bell, but is set to the n -gram order by default in LingPipe.

Bounded language models assume distinct begin-of-string (**BOS**) and end-of-string (**EOS**) string markers, setting $\hat{p}(\sigma) = \hat{p}(\sigma \text{ EOS} | \text{BOS})$, where the conditional probability is estimated using a process model. With string boundary padding, normalization is over all strings, with $\sum_{\sigma \in \Sigma^*} \hat{p}(\sigma) = 1$.

3 HMMs with Character Language Model Emissions

LingPipe employs first-order HMMs for tagging, where the hidden states, as usual, correspond to tags. Taggers assume a tokenization scheme that deterministically breaks an input into sequences of tokens. The joint probability of a token sequence $\sigma_1, \dots, \sigma_n$ and tag sequence t_1, \dots, t_n is defined by $p(\sigma_1, \dots, \sigma_n, t_1, \dots, t_n) = p(t_1, \dots, t_n) \cdot p(\sigma_1, \dots, \sigma_n | t_1, \dots, t_n)$. A first-order HMM defines $p(t_1, \dots, t_n) = p_{start}(t_1) \cdot \prod_{i>1} p(t_i | t_{i-1}) \cdot p_{end}(t_n)$; note the special estimates for start and end tags, which ensures the sum of all token/tag sequences is 1.

In typical HMMs, emissions are estimated as multinomials, with some kind of special handling for unseen tokens. LingPipe's HMMs are unusual in that they estimate the probability $p(\sigma|t)$ of the token σ given the tag t using bounded character language models, one for each tag t . This has the advantage of including general n -gram subword features within a fully generative probability model, as well as defining a proper probability model normalized over the infinite set of possible string emissions.

LingPipe's HMMs come with three decoders. The first is a standard Viterbi first-best decoder [4]. The second is a standard n -best decoder, which applies a Viterbi pass in a forward stage and then uses these as A* estimates to perform an exact backward search to iterate over an arbitrary number of unnormalized estimates of $p(t_1, \dots, t_n | \sigma_1, \dots, \sigma_n)$. The third decoder is a forward-backward decoder, which computes conditional probabilities of a tag given an input sequence [4].

Consider input tokens $\sigma_1, \dots, \sigma_n$. The forward value for a tag t and input position i is $\text{fwd}(t, i) = p(\sigma_1, \dots, \sigma_{i-1}, \text{tag}(i) = t)$, which is the probability of the first $i - 1$ input tokens resulting in the token σ_i at position i being assigned tag t . This value is estimated in linear time using the forward algorithm, at each stage computing the forward value as the sum of the values of all transitions from the previous forward values. Backward values for position i and tag t are defined by $\text{bk}(t, i) = p(\sigma_i, \dots, \sigma_n | \text{tag}(i) = t)$, the conditional probability of the current and remaining tokens given that the current tag is t . Backward probabilities are also easily computed in a single linear-time pass. Multiplying the forward and backward values produces the joint probability of a tag given an input sequence, $p(\sigma_1, \dots, \sigma_n, \text{tag}(i) = t) = \text{fwd}(t, i) \cdot \text{bk}(t, i)$. The conditional probability of position i receiving tag t is derived by marginalization, $p(\text{tag}(i) = t | \sigma_1, \dots, \sigma_n) = p(\sigma_1, \dots, \sigma_n, \text{tag}(i) = t) / \sum_{t'} p(\sigma_1, \dots, \sigma_n, \text{tag}(i) = t')$.

4 HMM Encodings for Chunking with Confidence

It is common to encode a chunking problem, such as named entity extraction, as a tagging problem. The typical tag set for a task like BioCreAtIvE would involve three tags: B_G for the first token in a gene mention, I_G for other tokens in a gene mention, and O for tags that are not part of a gene mention. It is possible to assign chunk probabilities with these tags, but the algorithm is tricky because of the lack of end markers [3]. This encoding is also problematic for our first-order HMMs; they tend to have difficulty finding boundaries, especially end boundaries.

We solve the search and estimation together using an encoding that is sensitive to position, using tags B_G (first token in mention), M_G (internal token in mention), E_G (last token in mention), and W_G (single token mention). Furthermore, we subcategorize the non-gene tags the same way (B_O , M_O , E_O and W_O). This distinguishes the first and last words in gene mentions, as well as the words directly preceding and following a gene mention.

With this encoding, the conditional probability of a subsequence of tokens being a gene mention given the entire sequence, $p(\sigma_i, \dots, \sigma_k : G | \sigma_1, \dots, \sigma_n)$, is:

$$\text{fwd}(B_G, i) \cdot \hat{p}(\sigma_i | B_G) \cdot \left(\prod_{i < j < k} \hat{p}(\sigma_j | M_G) \cdot \hat{p}(M_G | M_G) \right) \cdot \hat{p}(E_G | M_G) \cdot \text{bk}(E_G, k)$$

The probability of a single token gene mention is just the conditional tag probability, which is the product of the forward and backward estimates. LingPipe iterates the chunks in conditional probability order using an exact best-first search that keeps all partial entities on a priority queue, always expanding the one with highest probability, and popping and returning an answer when one is found.

5 Results on BioCreAtIvE II Gene Mention Data

LingPipe was trained on the BioCreAtIvE II data (see [5] and this volume), using default settings. Given the sentence *p53 regulates human insulin-like growth factor II gene expression through active P4 promoter in rhabdomyosarcoma cells*, the phrases extracted as chunks and their conditional probability estimates are *p53*: 0.999, *P4 promoter*: 0.733, *insulin-like growth factor II gene*: 0.606, *human insulin-like growth factor II gene*: 0.382, *active P4 promoter*: 0.140, *P4*: 0.092, *active P4*: 0.009, *insulin-like growth factor II*: 0.007, *human insulin-like growth factor II*: 0.004. The full precision versus recall curve is as follows:

	.02	.10	.20	.30	.40	.50	.60	.70	.80	.90	.95	.99	.999	.9999
Precision	.83	.76	.72	.69	.65	.61	.54	.46	.36	.25	.18	.11	.08	.07

This curve is computed by sorting all genes output in confidence order and then moving down the list, computing precision and recall at each point; average precision just averages precision values. For instance, LingPipe extracts 95% of all gene mentions in a list with 18% precision, and 99.99% of all mentions with 7% precision. Average precision is 55%. Average precision increases with our longer-distance rescorer models, but precision at 99.99% suffers, we suspect due to the increased variance and lowered bias. Overtraining helps on average, but hurts at the tail. We suspect discriminative models tuned for 0/1 loss would fare even worse.

References

- [1] Alias-i. 2006. LingPipe 2.3.0. <http://www.alias-i.com/lingpipe>. (BioCreAtIvE II in sandbox).
- [2] Carpenter, B. 2006. Character LMs for Chinese word segmentation and NER. *SIGHAN*.
- [3] Culotta, A. and A. McCallum. 2004. Confidence estimation for information extraction. *NAACL*.
- [4] Manning, C. and H. Schütze. 1999. *Found. of Stat. Natural Language Processing*. MIT Press.
- [5] Tanabe, L, N. Xie, L. H. Thom, W. Matten, and W. J. Wilbur. 2005. *BMC Bioinformatics*.



IASL Systems in the Gene Mention Tagging Task and Protein Interaction Article Sub-task

Hong-Jie Dai¹

hongjie@iis.sinica.edu.tw

Richard Tzong-Han Tsai^{1,*}

thtsai@iis.sinica.edu.tw

Hsi-Chuan Hung¹

yabt@iis.sinica.edu.tw

Wen-Lian Hsu^{1,*}

hsu@iis.sinica.edu.tw

¹ Institute of Information Science, Academia Sinica, 115 Nankang, Taipei, Taiwan

1 Overview

Named entity recognition (NER) is a crucial step for information extraction of relationships between genes and gene products. BioCreAtIvE II Gene Mention (GM) tagging task is concerned with this problem. The first part of this paper employs: 1) Conditional random fields (CRF) as the underlying machine learning model, 2) A set of features which are selected by sequential forward search algorithm, 3) Numerical normalization, and 4) Pattern-based post processing to resolve the GM task.

For GM task, we collect training/testing/development dataset from BioCreAtIvE I [1] and II to form a 15,443 sentences training set. In order to make use of this training set, we build a rule-based tokenizer based on the dataset from BioCreAtIvE I Task 1A. This tokenizer is also used to tokenize the training/testing set in our BioCreAtIvE II GM task and Protein Interaction Article Sub-task 1 (IAS).

The second part of this paper is about identifying protein-protein interaction (PPI) related biomedical abstracts. We propose a novel feature representation scheme, contextual-bag-of-words, to exploit named entity information. We further improve the performance by extracting reliable and informative instances from unlabeled and likely positive data to provide additional training data.

This paper is organized as follows. In Section 2 we describe our GM tagging system. In Section 3 we describe our PPI-text classification system. Finally, we conclude our work briefly in Section 4.

2 Gene Mention (GM) Tagging Task

Before describing our system, we first explain the way we used to formulate the NER problem. According to the IOB2 format, we transform the original sentence into a token/tag format. For example, the sentence “Comparison with alkaline phosphatases and 5-nucleotidase” will be transformed to “Comparison/O with/O alkaline/B phosphatases/I and/O 5-nucleotidase/B”.

2.1 System Description

After formulating the NER problem, we use seven feature types, including word, bracket, orthographical, part-of-speech (POS), affix, character-*n*-gram, and lexicon, to represent the characteristics of biomedical name entities (NEs). We explain them in the next section.

In order to leverage the performance and memory usage, we employ sequential forward selection (SFS) algorithm to find the best feature set and numerical normalization to reduce the number of features. Finally, we apply global patterns to fix the tag dependency outside the context window.

2.1.1 Feature Selection

It is inefficient to include all features in a Bio-NER model since memory resources are limited, and some features are ineffective. For our dataset, we divide it into a training set (10,298 sentences) and a development

* corresponding authors

set (5,153 sentences). Due to time and space limitations, it is very difficult to select a globally optimal feature set for the development set. We employ sequential forward selection algorithm to find the best feature set.

The algorithm is described as follows. We first calculate which feature has the highest F-score and select this feature as the basis for the feature pool. In each subsequent iteration, we individually add one feature type to the feature pool and calculate their F-scores, each time selecting the best scoring feature type and adding it to the pool. This process continues until the F-score stops increasing.

2.1.2 Numerical Normalization

In addition to selecting the efficient feature set that maximizes performance with limited memory resources, we also apply numerical normalization to reduce the number of features in each feature set. According to our observation, some proteins or genes of the same family usually differ in their numerical parts. For example, interleukin-2 and interleukin-3 belong to the same family—interleukin. In Bio-NER, they are both the target NE. Therefore, we normalize all numerals into one. For example, both interleukin-2 and interleukin-3 are normalized to interleukin-1.

2.1.3 Using Global Pattern to Improve CRF

The sequential tagging models we applied usually follows the Markov assumption that the current tag only depends on the previous tag. However, in Bio-NER, there are many exceptions. An NE may depend on the previous or next NE, or words among these NEs. Common sequential models cannot model this dependency. Furthermore, the sequential model only uses the information in the limited context window. It may fail if there are dependencies beyond the context window. To alleviate these problems, we apply global patterns composed of NEs and surrounding words.

Global Pattern Induction and Filtering

The first step in creating global patterns is to apply numerical normalization to all sentences in the training, development, and test sets. For each pair of sentences in the training set, we apply the Smith-Waterman local alignment algorithm [2] to find the longest common string, which is then added to the candidate pattern pool. During the alignment process, for each position, either of the two inputs that share the same word or NE can be counted as a match. The similarity function used in the Smith-Waterman algorithm is:

$$\text{Sim}(x, y) = \max \begin{cases} 1, & x = y \\ 1, & x's \text{ tag is } B \text{ or } I \text{ and } y's \text{ tag is } B \text{ or } I \\ 0, & \text{otherwise} \end{cases}$$

where x and y referred to any two compared tokens from the first and second input sentences, respectively. The similarity of two inputs is calculated by the Smith-Waterman algorithm based on this token-level similarity function.

Then we illustrate how patterns are extracted from a sentence pair in the training set. Given the following two tagged sentences:

...chemical/O interactions/O that/O **inhibit**/O butyrylcholinesterase/**B and**/O ...

and

...combinations/O of/O chemicals/O that/O **inhibit**/O butyrylcholinesterase/**B and**/O ...

, we will generate the "**inhibit <NE> and**" pattern. Here, we use bold face for the aligned words and tags in bold font. The first and last tokens in a pattern are constrained to be words, sentence beginning or ending symbols.

The extracted patterns are composed of a headword, NE type and a tail-word, e.g., "headword <NE type> tail-word." To test its effectiveness, each pattern is applied to the development set to correct the NE tags of all sentences. If the pattern's error ratio exceeds a certain threshold, τ , it is filtered out.

2.2 Feature Set

2.2.1 Word and bracket Features

Words preceding or following the target word may be useful for determining whether it is an NE or not. We use window size from -1 to 1, that is, the previous word, current word, and next word. We also include a feature to indicate whether the current token occurs within brackets or inside quotations.

2.2.2 Character-*n*-gram Features

A character *n*-gram is a substring of *n* characters of a longer string [3]. This feature helps our system to recognize NEs according to certain informative substrings, such as "ase" in "decarboxylase". In our system, we use character substrings of length 3 to 4 characters.

2.2.3 Orthographical Features

Table 1 lists all orthographical features used in our system. These features are widely used in other general NER [4] or biomedical NER systems [5].

Table 1: Orthographical features

Feature name	Regular Expression
INITCAP	^[A-Z].+
CAPWORD	^[A-Z][a-z]+\$
ALLCAPS	^[A-Z]+\$
CAPSMIX	^[A-z]*([A-Z][a-z][a-z][A-Z])[A-z]*\$
ALPHANUMMIX	^[A-z0-9]*([0-9][A-z][A-z][0-9])[A-z0-9]*\$
ALPHANUM	^[A-z]+[0-9]+\$
UPPERCHAR	^[A-Z]\$
LOWERCHAR	^[b-z]\$
SHORTNUM	^[0-9][0-9]?\$
INTEGER	^-[0-9]+\$
REAL	^-[0-9].[0-9]+\$
ROMAN	^-[IVX]+\$
HASDASH	-
INITDASH	^-
ENDDASH	-\$
PUNCTUATION	^[,;:?!]\$
QUOTE	^["'"]\$

2.2.4 POS Features

POS information is quite useful for identifying named entities. The GENIA POS tagger [6] and MEDPOST tagger [7] are used to provide POS information.

2.2.5 Affix Features

Affixes including prefixes and suffixes are morphemes. They are attached to base morphemes, such as roots, or to stems, to form words. Some of them can provide information to identify NE. For example, words ending in "~ase" are usually proteins. The length we used for prefixes and suffixes is 2-4 characters.

2.2.6 Lexicon Features

Finally, we include two kinds of lexicon features: exact match and dictionary distance. The first kind is just a binary feature indicating whether a token occurs in our lexicon or not.

In reality, it is difficult to find a lexicon which contains all possible variations of biomedical names. Therefore, it is useful to measure the distance between tokens and words in an external lexicon and set this as a feature. We use the Jaro-Winkler distance metric to compute the minimum distance between a token *x* and

an entity e in lexicon. These features are useful [8] because partial matches to entity names are informative. The lexicons we used are extracted from HUGO [9] and BioCreAtIvE I dataset.

2.3 Results

Table 2 shows the result of our three runs in BioCreAtIvE II test set. The best F-Measure is Run 3 which uses all seven feature types and applies post processing. We can see that adding lexicon features increases the precision of our system by 0.13%.

Table 2: Final results

Run ID	Run	Precision	Recall	F-Measure
1	No-lexicon feature	92.69%	68.73%	78.93%
2	With lexicon feature	92.82%	68.82%	79.04%
3	Post processing	92.67%	68.91%	79.05%

Table 3 shows the results of our system on the development set, which are relatively balanced in precisions and recalls in the development set. However, in the test set, our system achieves higher precisions but lower recalls. We believe that this is due to the strategy we used to create gold standard for the development set. Our development set is selected from training sets in BioCreAtIvE I and II. Some selected sentences exist in both BioCreAtIvE I and II datasets. These sentences are sometimes tagged differently in BioCreAtIvE I and II. We treat the BioCreAtIvE II annotation as the gold standard and BioCreAtIvE I as the alternative answers. Therefore, there may be many alternative answers for an NE in the development set. But in BioCreAtIvE II's test set, the gold standard was not created in this way. We believe that on average, the number of alternative answers per NE in the test set is less than that in the development set. This phenomenon causes the lower recalls in the test set.

Table 3: The performance on our development set

Run	Precision	Recall	F-Measure
No-lexicon feature	78.40%	81.75%	80.04%
With lexicon feature	78.86%	81.51%	80.17%

3 Protein Interaction Article Sub-task (IAS)

Before extracting PPI information from biomedical abstracts, it is necessary to identify them in the ever-increasing corpus of biomedical abstracts. This is the purpose of the BioCreAtIvE II IAS task. This task can be formulated as a text classification (TC) problem in the biomedical domain. We consider the following three critical issues in developing our PPI-TC system.

Adopting Contextual Information. In TC, documents are usually represented by bag-of-words (BoW) features. However, in PPI-TC, some words are informative only in certain contexts. For example, "bind" is more informative in indicating if an abstract is PPI-relevant when it appears in a sentence that has at least two proteins.

Filtering Out Likely Positive Instances. Gene Ontology (GO) is a widely used taxonomy that classifies many discovered protein interaction types, whereas a PPI database usually contains only some specific types that may not satisfy our requirements. Therefore, we usually treat abstracts annotated in PPI databases as likely positive (LP) examples. Those abstracts that do not contain PPI types of interest need to be filtered out.

Selecting Likely Negative Instances. It is easy to acquire a large number of positive (PPI-relevant) abstracts from PPI databases for use as LP data. On the other hand, likely-negative (LN) instances are often quite scarce. Since, most machine learning (ML) models used in classification require a balanced number of LP and LN examples, we must select more LN instances.

3.1 Method

3.1.1 Support Vector Machines and Term Weighting

The support vector machine (SVM) model is one of the best known ML models that can handle sparse high dimension data, which has been proved useful for text classification [10]. It tries to find a maximal-margin separating hyperplane $\langle \mathbf{w}, \phi(\mathbf{x}) \rangle - b = 0$ to separate the training instances, i.e.,

$$\min \|\mathbf{w}\|^2 + C \sum_i \xi^{(i)} \quad \text{subject to } y^{(i)}(\langle \mathbf{w}, \varphi(\mathbf{x}^{(i)}) \rangle - b) \geq 1 - \xi^{(i)}, \quad \forall i$$

where $\mathbf{x}^{(i)}$ is the i th training instance which is mapped into a high-dimension space by $\varphi(\cdot)$, $y_i \in \{1, -1\}$ is its label, $\xi^{(i)}$ denotes its training error, and C is the cost factor (penalty of the misclassified data). The mapping function $\varphi(\cdot)$ and the cost factor C are the main parameters of a SVM model.

When classifying an instance \mathbf{x} , the decision function $f(\mathbf{x})$ indicates that \mathbf{x} is "above" or "below" the hyperplane. [11] shows that the $f(\mathbf{x})$ can be converted into an equivalent dual form which can be more easily computed:

primal form: $f(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \varphi(\mathbf{x}) \rangle - b)$; dual form: $f(\mathbf{x}) = \text{sign}\left(\sum_i \alpha^{(i)} y^{(i)} K(\mathbf{x}^{(i)}, \mathbf{x}) - b\right)$

where $K(\mathbf{x}^{(i)}, \mathbf{x}) = \langle \varphi(\mathbf{x}^{(i)}), \varphi(\mathbf{x}) \rangle$ is the kernel function and $\alpha^{(i)}$ can be thought of as w 's transformation.

In the IAS subtask, we chose the following polynomial kernel according to our preliminary experiment results:

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = (\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle + 1)^2 \text{ and } C = 1$$

In the text categorization problem, a document d is usually represented as a term vector \mathbf{v} . Each dimension v_i in \mathbf{v} corresponds to a term t_i . v_i is calculated by a term weighting function, which is very important to SVM-based text categorization because SVM models are sensitive to the data scale, namely dominated by some widest dimensions. In this paper, we employ the three most popular functions: Binary, TFIDF, and BM25, which are defined as follows:

$$\text{Binary}(t_i, d) = \begin{cases} 1 & \text{if } t_i \text{ appears in } d \\ 0 & \text{otherwise} \end{cases},$$

$$\text{TFIDF}(t_i, d) = \text{TF}(t_i, d) \cdot \text{IDF}(t_i, D).$$

where D is the document set that contains all documents in the training and test sets,

$$\text{TF}(t_i, d) = \frac{t_i \text{'s frequency in } d}{\text{word counts of } d}, \text{ and } \text{IDF}(t_i, D) = \frac{\#\text{ documents } \in D \text{ containing } t_i}{|D|}$$

BM25's definition of can be found in [12].

3.1.2 Methods of Exploiting Named Entity Information

A PPI abstract must contain some protein names. Hence, recognition of protein names in abstracts can improve the identification of PPI abstracts. We use our GM tagging system to provide NEs information. In the following we describe our new feature representation scheme.

Contextual Bag of Words (CBoW). The number of protein names that exist in the context affects a word's informativeness for PPI relevance. Based on this fact, we distinguish the original word bags into different contextual bags. The words in individual sentences are bagged according to the number of protein named entities (NEs) in the sentence. If there are 0 NEs the words are put into contextual bag 0; if 1 NE, then bag 1; and if 2 or more NEs, then bag 2.

For comparison, we implement two well-known features that should be incorporated with BoW features:

Bag of Phrases (BoP). [13] suggested that adding phrases into the original bags can retain some order information which is lost in BoW. In our case, we add protein NE phrases into bags.

Bag of Normalized NEs (BoN). The more protein names that appear in an abstract, the more likely it is to be PPI-relevant. Following [14], we replace each NE in a given abstract with “PROTEIN_{*i*}”, where *i* denotes the order of appearance in this abstract. Abstracts containing different numbers of NEs have different normalized NE features.

3.1.3 Filtering Out Likely-Positive Instances and Selecting Likely-Negative Instances

To filter out irrelevant data from likely-positive data, we use the initial model that is trained on TP+TN using only BoW features. Those abstracts in the original LP with an SVM output in $[\gamma^+, 1]$ are retained, where γ^+ is chosen to be 0. The dataset produced by filtering out irrelevant LPs is referred to as selected likely positive data (LP*).

To select likely negative instances, we employ a bootstrapping-like technique inspired by [15]. We collect 50k unlabeled abstracts from the PubMed biomedical literature database and classify them with our initial model. The articles with an SVM output in $[-1, \gamma^-]$ form the selected likely-negative (LN*) dataset, where $-1 < \gamma^- < 0$ is a threshold. γ^- is chosen to be -0.9. The articles with predicted values less than -1 are excluded since they are absolutely negative examples that may not be useful for determining the hyperplane in SVM. In addition, the instances whose SVM outputs are in $[\gamma^-, 0]$ are discarded due to unreliability.

3.2 Results

Three datasets provided by BioCreAtIvE II are shown in Table 4. For each abstract, we remove all punctuation symbols, numbers, and stop words in the preprocessing step. We use our GM tagging system to tag NEs in each abstract. Before applying our system to the test set from BioCreAtIvE II IAS task, we conduct 10-fold cross validation experiments on the training set and use the F-Measures to score our system.

Table 4: Three datasets in IAS

Dataset	Size
True positive (TP)	3536 abstracts
True negative (TN)	1959 abstracts
Likely-positive (LP)	18930 abstracts

3.2.1 Exploiting Named Entity Information

Table 5 shows the 10-fold cross validation results on the training set for different IAS methods that exploit NE information. CBoW appears to outperform BoW, whereas the other two configurations that incorporate NE features into BoW only slightly improve the performance of BoW regardless of the weighting.

Table 5: F-Measures of different IAS methods of using NEs

Features	binary	TF-IDF	BM25
BoW	93.85	94.04	94.41
BoW + BoP	94.01	94.15	94.47
BoW + BoN	94.71	94.92	94.70
CBoW	95.85	96.01	97.34

3.2.2 Expanding the Training Set

In this section, we examine the effects of adding LP* and LN*. Without loss of generality, we use the CBoW representation scheme. As shown in Table 6, adding the selected data slightly improves the F-Measure of all weight schemes.

Table 6: F-Measures of original training set vs. the expanded one

Configuration	binary	TF-IDF	BM25
TN+TP	95.85	96.01	97.34
TN+TP+LN*+LP*	96.16	96.18	97.91

3.2.3 Results of IAS Task

Table 7 shows the results on the test set, including our IAS system’s performance along with the mean and

median scores of all the participant systems. Our Run1 system employs the best feature set found in the development set. It uses the LP* and LN* data while our Run2 system does not. We can see that with LP* and LN*, the performance can be slightly improved by 1.10%. These results are similar to those in Table 6. In addition, both Run 1 and 2 significantly outperform the mean and median scores. This shows that our CBoW representation is generally effective in the IAS task.

Table 7: Performance on the test set

Configuration	Precision	Recall	F-Measure
Run 1 (TN+TP+LN*+LP*)	68.90%	85.07%	76.13%
Run 2 (TN+TP)	66.46%	86.13%	75.03%
Mean	66.42%	76.36%	68.68%
Median	67.72%	85.07%	72.24%

4 Conclusions

In the work, we first propose a NER system in the biomedical domain using SFS feature selection and numerical normalization to efficiently utilize the memory and maximize tagging performance. We use the Smith-Waterman local alignment algorithm to help ML-based Bio-NER to deal with extremely difficult cases which need longer context windows.

Finally, we propose a novel CBoW feature representation scheme and demonstrate that it outperforms other methods that also exploit NE information in PPI-TC. We also extract likely positive and likely negative data for enhancing the performance of PPI-TC. Our study of the PPI-TC problem presents a potential new direction of exploiting NLP-based contextual information.

References

- [1] Hirschman, L., Yeh, A., Blaschke, C., and Valencia, A., Overview of BioCreAtIVe: critical assessment of information extraction for biology, *BMC Bioinformatics*, 2005.
 - [2] Smith, T.F. and Waterman, M.S., Identification of common molecular subsequences, *J. Mol. Biol.*, 147(1):195-197, 1981.
 - [3] Cavnar, W.B., Using an n -gram-based document representation with a vector processing retrieval model, *Proc. TREC-3 Conference*, 269–278, 1994.
 - [4] Florian, R., Ittycheriah, A., Jing, H., and Zhang, T., Named entity recognition through classifier combination, *Proc. CoNLL-03 Conference*, 168-171, 2003.
 - [5] Zhou, G. and Su, J., Exploring deep knowledge resources in biomedical name recognition. *Proc. JNLPBA-04 Conference*, 2004.
 - [6] Tsuruoka, Y. and Jun'ichi Tsujii, J., Bidirectional inference with the easiest-first strategy for tagging sequence data. *Proc. HLT/EMNLP-05 Conference*, 2005.
 - [7] Smith, L., Rindflesch, T., and Wilbur, W.J., MedPost: a part-of-speech tagger for bioMedical text. *Proc. Bioinformatics*, 2004.
 - [8] Cohen, W.W. and Sarawagi, S., Semi-markov conditional random fields for information extraction, *Proc. NIPS-04 Conference*, 2004.
 - [9] <http://www.gene.ucl.ac.uk/nomenclature/>
 - [10] Joachims, T., Text categorization with support vector machines: learning with many relevant features. *Proc. ECML-98. Conference*, 1998.
 - [11] Cristianini, N. and Shawe-Taylor, J., *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.

- [12] Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M., Okapi at TREC-3, *Proc. TREC-3 Conference*, 1994.
- [13] Scott, S. and Matwin, S., Feature engineering for text classification, *Proc. of ICML-99 Conference*, 1999.
- [14] Paradis, F. and Nie, J.-Y., Filtering contents with bigrams and named entities to improve text classification, *Proc. of AIRS-05 Symp.*, 2005.
- [15] Liu, B., Lee, W.-S., Yu, P., and Li, X., Partially supervised classification of text documents, *Proc. ICML-02 Conference*, 2002.



A Study for Application of Discriminative Models in Biomedical Literature Mining

Chengjie Sun Lei Lin Xiaolong Wang Yi Guan
`{cjsun, linl, wangxl, guanyi}@insun.hit.edu.cn`

Language Technology Center, School of Computer Science, Harbin Institute of Technology, No. 92 West Da-Zhi Street, Harbin, 150001, China

Abstract

By automatically identifying gene and protein names and mapping these to unique database identifiers, it becomes possible to extract and integrate information from a large amount of biomedical literature. This paper presents the attempts of use discriminative models to automatically detect gene name mention and normalize gene mentions. Conditional Random Fields model is adopted to solve gene mention task and Maximum Entropy model is used to do gene mention normalization task. The evaluation results of biocreative2006 are also reported.

Keywords: discriminative model, conditional random field, maximum entropy, text mining

1 Introduction

Biomedical literature contains significant parts of biological knowledge, but it is hard to integrate and maintain such knowledge due to the free format of biomedical literature. The explosion of literature in the biomedical field has provided an opportunity for natural language processing techniques to aid researchers and curators of databases in the biological field by providing text mining services. The discriminative models have been widely used in natural language processing field due to the good performance and the ability to combine heterogeneous features [1]. In biocreative2004 and JNLPBA2004, discriminative models, such as Maximum Entropy, Maximum Entropy Markov model and Conditional Random Fields, also have been widely adopted.

In this paper, we detail our methods for Gene Mention (GM) task and Gene Normalization (GN) task of biocreative2006 and report our results. Section 2 and section 3 describe the methods and evaluation results for the two tasks respectively.

2 Gene Mention Task

2.1 Method for GM Task

The GM task could be addressed a sequence labeling problem. In practice, we regard each word in a sentence as a token and each token is associated with a label. Each label with a form of B-C, I-C or O indicates not only the category of a gene name but also the location of the token within the name. In this label denotation, C is the category label; B and I are location labels, standing for the beginning of a name and inside of a name. O indicates that a token is not part of a name. For GM task, there is only one category, so we have 3 labels all together: B-gene, I-gene and O.

In our system, we utilize Conditional Random Fields model, which is a discriminative model and very suitable to sequence labeling problem, to solve GM task. Features are vital to the system performance. Our feature types include orthographical features, context features, word shape features, prefix and suffix features, Part of Speech (POS) features and shallow syntactic features. POS tags and shallow syntactic (chunking) tags are gotten by using GENIA Tagger [2]. Experiments show that our method can achieve an F-measure of 71.2% in JNLPBA2004 test data and which is better than most of state-of-the-art systems.

The meaning of each feature type is listed as following, where “c” denotes a chunk label, “p” denotes a POS label, “w” denotes a token, -n denotes n position prior to target token and +n denotes n position after target token.

1) Orthographical Features

Table 1: Orthographical features

Feature name	Regular Expression
ALLCAPS	[A-Z]+
INITCAP	^[A-Z].*
CAPSMIX	.*[A-Z][a-z].*[a-z][A-Z].*
SINGLE CHAR	[A-Za-z]
HAS DIGIT	.*[0-9].*
SINGLE DIGIT	[0-9]
DOUBLE DIGIT	[0-9][0-9]
NATURAL NUMBER	[0-9]+
REAL NUMBER	[-0-9]+[.][0-9,.]+
HAS DASH	.-*
INIT DASH	-.*
END DASH	*-
ALPHA NUMERIC	(.*[A-Za-z].*[0-9].*)(.*[0-9].*[A-Za-z].*)
ROMAN	[IVXDLCM]+
PUNCTUATION	[.,;?!-+]

2) Context Features:

w-2, w-1, w0, w1, w2, w-1w0, w0w1

3) Part-of-speech Features:

p-2, p-1, p0, p1, p2, p-1p0, p0p1, p-1p0p1

4) Word Shape Features:

Kappa-B =>XXXXX_X

Kappa-B =>Xx_X

5) Prefix and Suffix Feature:

Length with 3, 4, 5 both for prefix and suffix

6) Chunk features:

c-2, c-1, c0, c1, c2

7) Combine feature

p-1c0, c0t0 and p0c0

2.2 Results of GM Task

We use the CRF tool in Mallet toolkit [3] to train the model on the given training data. No other resource or data are involved. We submitted two runs for GM task in biocreative2006. The difference between them is that run2 uses the stemmed token while run1 uses the raw token. The results are shown in Table 2.

From Table 2, we can see that stemming isn't helpful in GM task. Our system's performance is comparable to what we got from JNLPBA2004 test data, but the performance is relative low in biocreative2006. This is maybe caused by the difference between the two corpora. Also, our system doesn't involve biomedical resources such as dictionary or ontology, which also could decrease the system's performance.

Table 2: Biocreative2006 results of GM task

	Precision (%)	Recall (%)	F-measure (%)
Run1	80.46	73.61	76.88
Run2	80.81	72.48	76.42

3 Gene Normalization Task

3.1 Method for GN Task

Inspired by [4], we build a model that, given a set of synonym matches, distinguishes correct from incorrect ones. This is essentially a binary classifier in which good matches are positively labeled and bad matches negatively labeled. Another discriminative model, Maximum Entropy (ME) model is chosen to do the classification task in our system.

To create training data for the classifier, we matched every synonym (in entrezGeneLexicon.list file) to each training document using a strict literal matching criterion (loose matching criterion may be better. We didn't use this strategy because of time limited). We then extracted, for each match, the text that matched, the three words right before the match, the three words right after the match ([4] used two words both before and after the match) and the normal form causing the match. For the training data, if the normal form for a match was in the normalized gene list for that document, then the match was labeled positive; otherwise, it was labeled negative. This provided a large set of positive and negative matches required to train an ME classifier.

To classify a new abstract, the system first extracts all the synonym matches that occur within it. Then for each match, the classifier will judge whether it is positive or negative. For one positive match, the normal form causing the match is added to the documents normalized gene list.

3.2 Results of GN Task

Zhangle's ME tool [5] is used to train the ME model. For GN task, we submitted 3 runs to biocreative2006. For run1, we didn't consider the matched protein name as a whole when doing features collection in the ME model. For example "SYT protein" was treated as two tokens. For run2, we considered the matched protein name as a whole when doing features collection in ME model. For example "SYT protein" was treated as one token "SYT_protein". Run1 returns more results than run2 and both of them didn't involve noisy training data. For run3, we added about 10,000 matches found in noisy training data into the training matches and the protein name was considered as a whole as in run2. The evaluation results are shown in Table 3.

Table 3: Biocreaticve2006 results of GN task

	Precision	Recall	F-measure	True Positive	False Positive	False Negative
Run1	0.361	0.429	0.392	337	596	448
Run2	0.375	0.415	0.394	326	543	459
Run3	0.419	0.331	0.370	260	361	525

From Table 3, we can conclude that simply involving more training examples from noisy training data couldn't prompt the system's performance. An active learning strategy may solve this problem. Also, treating a protein name as a whole token is helpful. Besides, loosening the matching criterion can afford more positive training instances, which also should be good for performance promotion according to [4].

Acknowledgements. We extend our special thanks to Minghui Li, Qiang Wang and Yancheng He for their help during the system development. This work is supported by National Natural Science Foundation of China (60435020 and 60673019).

References

- [1]Lafferty, J., McCallum, A. and Pereira, F., Conditional Random Fields, Probabilistic Models for Segmenting and Labeling Sequence Data, In Proceedings of the International Conference on Machine Learning, 282–289, 2001
 - [2]Tsuruoka, Y., Tateishi, Y., Kim, J.D., Developing a Robust Part-of-Speech Tagger for Biomedical Text. In Advances in Informatics - 10th Panhellenic Conference on Informatics, 382-392, 2005.
 - [3]McCallum, A., MALLET: A Machine Learning for Language Toolkit. <http://mallet.cs.umass.edu>, 2002.
 - [4]Crim, J., McDonald, R. and Pereira, F., Automatically annotating documents with normalized gene lists, BMC Bioinformatics 2005, 6:S13, 2005.
 - [5]<http://homepages.inf.ed.ac.uk/s0450736/maxent.html>



Fundación
Centro Nacional
de Investigaciones
Oncológicas Carlos III

Melchor Fernández Almagro, 3
28029 Madrid
Tel. +34 91 224 69 00
Fax +34 91 224 69 80
www.cnio.es



ISBN: 84-933255-6-2